computation

MDPI

Article

A Study on Locally Runnable Large Language Models for Bearing Fault Diagnosis

Mehadi Hasan Shawon and Prashant Kumar *

Department of AI and Big Data, Woosong University, Daejeon 34606, Republic of Korea; shawon@iwmi.kr
* Correspondence: prashant@wsu.ac.kr

Abstract

Large language model (LLM) agents can perform prognostics and health management (PHM) tasks such as bearing fault diagnosis, but most published systems rely on large, paid, cloud-hosted models that a small or medium enterprise (SME) cannot self-host. The diagnostic accuracy that can be achieved on free, offline, commodity hardware is a practical question. Using an execution-based evaluation (Pass@1, Pass all 3, macro F1), this paper benchmarks eight small, publicly accessible, locally runnable LLMs (1B–9B parameters, via Ollama) on vibration-derived features for three-class bearing fault diagnosis (Healthy, Outer race, and Inner race). The proposed work is evaluated on three independent datasets, Paderborn, CWRU, and HUST, across a 0–5-shot ablation and at two decoding temperatures to separate accuracy from reliability. The findings replicate across all three datasets, namely, a model-capability gate that only models near 7B parameters and above clears the majority-class floor; few-shot prompting is non-monotonic; accuracy and reliability are distinct axes; and a single free model, gemma2:9b (5.4 GB), is the most accurate and among the most reliable, with no task-specific training. We further propose ensemble agreement gating, a training-free reliability rule that withholds predictions when several free local models disagree, raising accuracy on the answered subset (e.g., 0.77 → 0.87 on Paderborn). In a head-to-head on identical prompts and data, the free local models match a current frontier model in the settings tested at zero cost and fully offline, and we characterize where such a training-free local approach is and is not appropriate for resource-constrained operators of rotating machinery. On the same features, however, a simple supervised baseline such as logistic regression, and even an untrained physics rule, outperform all eight LLMs, so we present this as a cautionary benchmark: the contribution of the free local approach is training-free deployment and a reliability gating rule, not classification accuracy.

Keywords: bearing fault diagnosis; large language models; local LLM; prognostics and health management; few-shot prompting; edge AI

Academic Editors: Xueyi Li, Tianyang Wang and Haochen Qi

Received: 26 August 2026
Revised: 13 September 2026
Accepted: 16 September 2026
Published: 19 September 2026

1. Introduction

Rolling-element bearings are among the most failure-prone components in rotating machinery, and their unexpected failure is a leading cause of unplanned downtime in pumps, motors, gearboxes, and turbines. Condition monitoring of these bearings is therefore one of the most common and economically important tasks in the prognostics and health management (PHM) of rotating machinery [1]. The dominant diagnostic signal is vibration: characteristic defect frequencies such as the ball pass frequencies of the inner and outer race, the ball spin frequency, and the fundamental train frequency appear in the

envelope spectrum as a localized fault develops, and a large body of signal-processing and machine-learning (ML) work has been built on top of these features [2].

Classical data-driven approaches train a supervised model, such as a support vector machine, a random forest, or a one-dimensional convolutional neural network (CNN), on labeled vibration data [3]. These methods achieve near-perfect in-distribution accuracy and remain the state of the art when abundant labeled data are available. However, they carry two practical costs that are easy to overlook in a laboratory setting: they require labeled fault data for each machine or operating condition, and they transfer poorly across rigs, so a model trained on one test bench typically must be retrained, retuned, and revalidated before it can be trusted on another. For an organization without a dedicated ML team, standing up and maintaining such a pipeline for every machine is rarely feasible.

More recently, large language models (LLMs) have been applied to fault diagnosis, and two lines of work have shown that they can reach competitive accuracy. In the first, LLM agents perform diagnosis by reasoning over extracted signal features within a scenario-driven, tool-using loop [4]. In the second, a task-specific fine-tuned Qwen2.5 7B model reaches 96.5% on a motor fault diagnosis benchmark [5]. More broadly, LLMs and large-scale foundation models are being adopted rapidly across prognostics and health management [6], with recent work bringing multimodal and spectral-feature LLMs to bearing fault diagnosis [7], proposing unified health management frameworks for rotating machinery [8], and developing autonomous LLM agents for predictive maintenance [9]. Both approaches are encouraging, but the systems that succeed do so under conditions that reintroduce the very barriers classical ML imposes: the agentic route depends on frontier, paid, cloud-hosted models [4], while the fine-tuning route requires task-specific training on labeled target data [5]. Each therefore assumes access to either a recurring cloud budget or a labeled dataset plus the in-house expertise to fine-tune, resources that a small or medium enterprise (SME) typically lacks. This exposes a clear research gap. On one side, supervised pipelines need per-machine labeled data and ML staff; on the other, the LLM systems that work need either a paid cloud model or fine-tuning on labeled data. What remains untested is the middle ground that is actually attractive to an SME: whether general-purpose, openly available, locally runnable LLMs, used out-of-the-box with no fine-tuning, no labeling, and no per-dataset training, can diagnose bearing faults from standard vibration features accurately enough to be useful, whether the same model and prompt generalize across different test rigs, and how a practitioner can know when to trust a prediction without any training signal. To our knowledge, no prior study characterizes free, local LLMs on this task across multiple datasets or provides a training-free reliability mechanism for them. For an SME, the frontier-model route is often a nonstarter: recurring per-call cost, data governance and air-gap requirements, and the absence of in-house ML staff all push toward a self-hosted, free, offline solution. This motivates the central question of this paper: which openly available, locally runnable LLMs, if any, are good enough to diagnose bearing faults from vibration features on commodity hardware, and how reliable are they? We answer it with a controlled benchmark of eight free, local models (1B–9B parameters) on three standard datasets: Paderborn [10], CWRU [11], and HUST [12], under a single fixed, leak-free protocol, with reliability estimated by self-consistency across repeated runs, adapting the execution-based pass@k idea from code-generation evaluation [13].

Recent work on reliable and data-efficient industrial AI is also relevant to this setting. Explainable and graph neural networks have been surveyed for process fault detection and diagnosis [14], adaptive graph neural networks have been combined with Kolmogorov-Arnold networks for industrial soft sensors [15], and reflective large language model inference has been explored for few-shot soft sensing under operating-condition drift [16]. For rotating machinery specifically, hybrid model-based schemes for wind turbines combine

neuro-fuzzy Takagi-Sugeno models with interval observers [17], and structural analysis has been paired with Gaussian-process interval estimation for fault detection [18]. These directions are complementary to the present study, which instead asks how far untuned, locally runnable general-purpose LLMs can go on standard bearing features. Concretely, the study pursues five objectives: (i) to determine whether, and which, locally runnable LLMs are accurate enough to be useful for bearing fault diagnosis without any task-specific training; (ii) to test whether a single model and prompt transfer unchanged across three independent datasets; (iii) to characterize reliability and derive a trust signal that does not depend on labeled data; (iv) to situate the free/local approach against both a supervised baseline and a current frontier model evaluated on identical prompts and data; and (v) to distill practical deployment guidance, namely a model-size gate, a few-shot prompt budget, and a reliability threshold, for practitioners.

The main contributions of the paper are listed below:

Training-free, expertise-free diagnosis approach: general-purpose free/local LLMs (1B–9B) diagnose bearing faults from standard vibration features with no per-dataset training, labeling, or tuning, removing the ML staff barrier that keeps supervised pipelines out of reach for SMEs.
Out-of-the-box cross-setup generality: the same model and prompt transfer unchanged across Paderborn, CWRU, and HUST and remain useful.
Practical deployment guidance: a model-size gate, a few-shot ablation, and a reliability axis (Pass all 3/self-consistency) that flags which models can be trusted.
A contextual comparison to supervised baselines and to a current frontier model on identical prompts and data, situating the free/local approach between "needs labeled data plus ML expertise per machine" and "needs a paid cloud model."
A training-free reliability method: *ensemble agreement gating* that combines several free local models by majority vote and abstains when they disagree, giving a deployable trust signal (selective prediction) at no additional training or labeling cost.

An honest baseline comparison: on the same features and the same test windows, a logistic regression, a shallow decision tree, and a random forest all outperform every tested local LLM, so this paper is a cautionary benchmark whose methodological contribution is the training-free reliability gating rather than diagnostic accuracy.

2. Datasets and Methods

2.1. Datasets

We evaluate on three widely used, independently collected bearing datasets Paderborn [10], CWRU [11] and HUST [12] each cast to the same three-class task (Healthy/Outer race/Inner race). CWRU ball fault records and HUST Ball, combination, and variable-speed records are excluded so the label space matches exactly. Splits are performed at the recording (file) level so that windows from one recording never appear in both the few-shot reference pool and the test set. The LLM test set is a fixed, seeded subset per dataset (150/150/140 windows for Paderborn/CWRU/HUST). Dataset parameters are summarized in Table 1. The full held-out test pools after the file-level split contain 542, 240, and 200 windows for Paderborn, CWRU, and HUST. To keep repeated LLM inference tractable, the LLMs are evaluated on a fixed subset of up to 50 windows per class, drawn with a fixed random seed independently of any model output: 150 windows for Paderborn (50/50/50), 150 for CWRU (50/50/50), and 140 for HUST (50/50/40, because the HUST healthy class has fewer test-pool files). All reported LLM numbers use these subsets.

Table 1. Datasets, all recast to the same 3-class task (Healthy/Outer/Inner).

Dataset	Signal	Sample Rate	Window	Classes	Windows (Ref/Test)
Paderborn (real, N15_M07_F10)	vibration_1	64 kHz	1.0 s	H/O/I	813/542
CWRU (12k drive-end)	DE accel.	12 kHz	1.0 s	H/O/I	310/240
HUST (constant-speed)	X accel.	25.6 kHz	1.0 s	H/O/I	300/200

2.2. Features

Each window is reduced to a compact, physically interpretable feature vector rendered as text: time-domain statistics (RMS, kurtosis, crest/impulse/clearance factors, skewness), spectral summaries, and envelope-spectrum amplitudes at the bearing characteristic defect frequencies (BPFO, BPFI, FTF, BSF and harmonics), computed from each dataset's own bearing geometry (Paderborn 6203; CWRU 6205; HUST ER 16K, 9 balls) and shaft speed. The BPFO-to-BPFI energy ratio cleanly separates the classes on all three datasets (CWRU means: Healthy 1.15, Inner 0.85, Outer 5.35), confirming the pipeline is physically correct. The complete feature-generation pipeline and a worked example are provided in Appendix A.

2.3. Models and Prompt

We test eight models runnable on an 18 GB Apple silicon laptop via Ollama: llama3.2:1b, gemma2:2b, qwen2.5:3b, llama3.2:3b, phi3.5, mistral:7b, qwen2.5:7b, and gemma2:9b (1.3–5.4 GB on disk) [19–24]. A neutral system prompt states textbook bearing physics only, with no data-derived thresholds, and asks for a single class label with a brief justification. Few-shot examples (0–5 per class) are drawn from the reference split and interleaved round-robin to avoid recency bias.

2.4. Protocol and Metrics

We use an execution-based evaluation: a prediction is graded correct only if it is a valid canonical label *and* an exact match to the ground-truth class. We report Pass@1 (correct on the first run) and Pass all 3 (correct on all three reruns), adapted from the execution-based pass@k evaluation of code-generation models [13] together with macro F1. This follows the execution-based philosophy of the agentic PHM benchmark PHMForge [4], which grades whether an agent's output meets task-validation criteria (reported there as a task-completion rate). We report temperature 0 for the deterministic headline and temperature 0.7 (three reruns) for reliability, since at temperature 0 the reruns are identical and Pass all 3 collapses to Pass@1. Temperature 0.7 is available for all three datasets and is therefore the setting used for every cross-dataset comparison in this paper, whereas the deterministic temperature 0 headline is reported for Paderborn; because sampling at 0.7 can only match or fall below greedy (temperature 0) accuracy, the temperature 0.7 figures are conservative. The k-shot ablation covers 0–5 examples per class. Every experiment is autosaved and resumable. We distinguish the zero-shot and few-shot settings. In the zero-shot setting, the model receives no labeled examples and the method is fully label-free; in the k-shot setting, a few labeled reference windows are placed in the prompt, so it is not label-free. Throughout, training-free means no fine-tuning, no parameter updates, and no per-machine model training: the few-shot examples are a handful of labeled windows in the context, not a training set.

2.5. Ensemble Agreement Gating (Proposed Method)

Because an SME must act on predictions it cannot independently verify, we propose a simple, training-free reliability mechanism that turns disagreement among several free models into a trust signal. Given M locally runnable models (here M = 3: gemma2:9b,

qwen2.5:7b, mistral:7b), each labels a window independently; the ensemble returns the majority label and an *agreement level* $a \in \{1, \ldots, M\}$ equal to the size of the winning vote. Predictions are then gated: windows below an agreement threshold (e.g., not unanimous) are withheld and routed to a human inspector, producing a selective-prediction (risk–coverage) trade-off. The rule needs no training and no labels, only additional inference on the same commodity hardware and is evaluated in Section 3.8.

3. Results

3.1. Headline

The best single configuration is qwen2.5:7b at 4-shot, 0.78 Pass@1/0.78 macro F1 on Paderborn at temperature 0, on hardware an SME already owns. At temperature 0.7, the common setting we use for every cross-dataset comparison, the best local model reaches 0.87 Pass@1 on CWRU (gemma2:9b) and 0.70 on HUST (gemma2:9b and mistral:7b); as noted in the Methods, these temperature 0.7 figures are conservative relative to greedy decoding. For context, the agentic benchmark PHMForge reports [4] its strongest paid, *cloud-hosted* frontier agent reaching 68% overall task completion (73.3% on its fault-classification scenarios); the tasks are not identical, so we do not treat this as a controlled comparison. Our like-for-like test against a frontier model on the same data and prompt is reported in Section 3.9, where the free local models show no accuracy gap. Figures 1 and 2 give the full Pass@1/Pass all 3 grids.

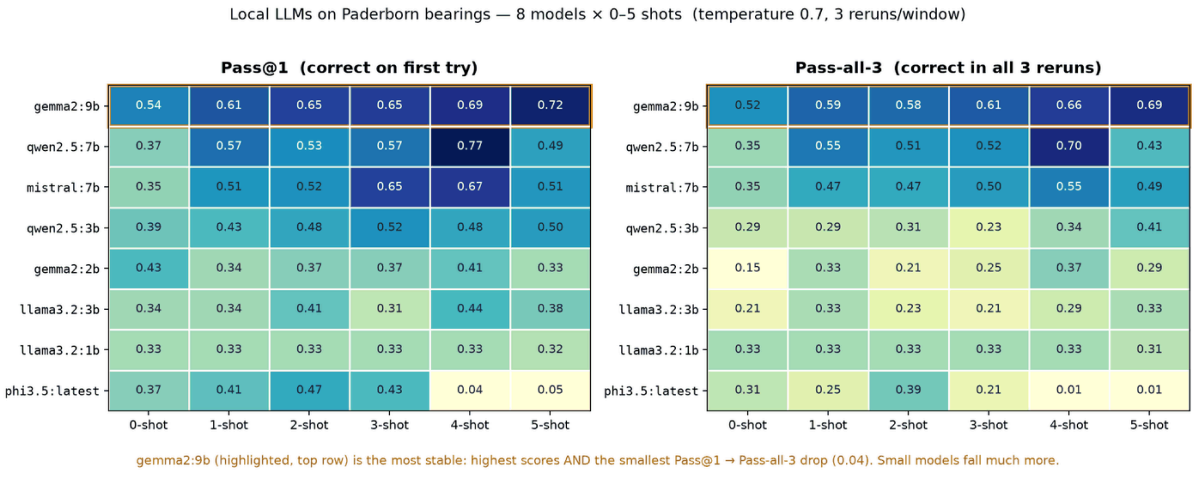

Figure 1. Paderborn: Pass@1 and Pass all 3 across 0–5 shots (temperature 0.7).

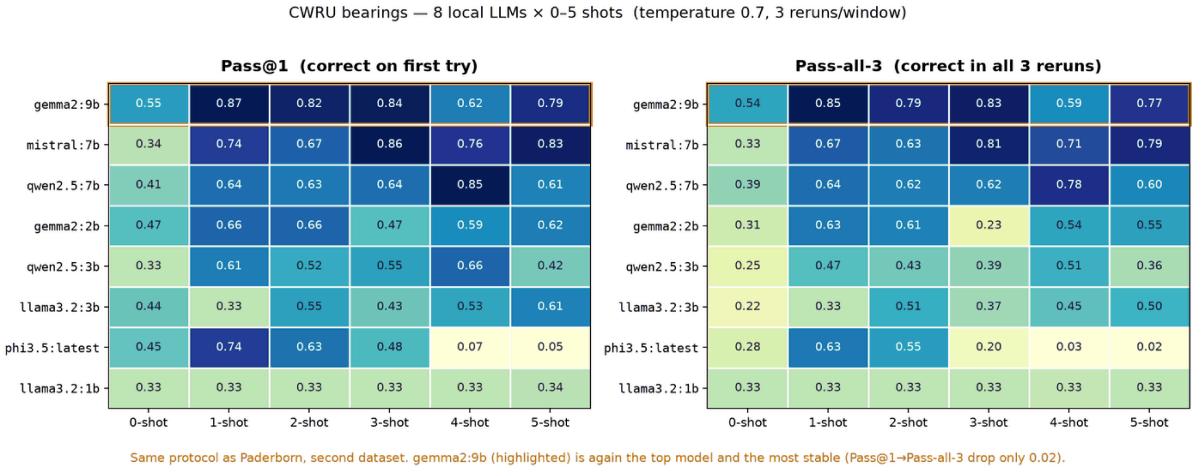

Figure 2. CWRU: same protocol, second dataset.

3.2. Per-Class Behavior

Table 2 reports per-class F1 at the 4-shot operating point on Paderborn, and Figure 3 shows representative confusion matrices. Capable models spread their errors between the two fault classes (Outer/Inner), which share an envelope-spectrum structure, whereas the collapsed llama3.2:1b assigns a single class to every window visually, an all "Healthy" column, exactly the "consistent but wrong" failure that a single accuracy number hides. The overlap between the inner- and outer-race signatures is expected, because both produce periodic impacts whose harmonics fall close together in the envelope spectrum, so a competent model's residual error is mostly confusion between the two fault types rather than a failure to notice a fault at all. By contrast, llama3.2:1b earns a per-class F1 of 0.5 for Healthy and 0.0 for both fault classes, which confirms that its 0.33 accuracy reflects the absence of a diagnosis rather than a weak one. Reading performance class by class, rather than as a single accuracy number, is what separates a genuine diagnosis from a lucky guess. The capable models are strongest on the Healthy class and weakest on the Inner-race fault: gemma2:9b, for example, reaches a per-class F1 of about 0.73 for Healthy, 0.73 for Outer, and 0.60 for Inner at this operating point.

Table 2. Per-class F1 at 4-shot on Paderborn (temperature 0). H/O/I = Healthy/Outer/Inner.

Model	Pass@1	$F1_H$	$F1_O$	$F1_I$
qwen2.5:7b	0.780	0.79	0.82	0.73
mistral:7b	0.733	0.80	0.68	0.70
gemma2:9b	0.680	0.73	0.73	0.60
qwen2.5:3b	0.493	0.57	0.20	0.59
llama3.2:1b	0.333	0.50	0.00	0.00

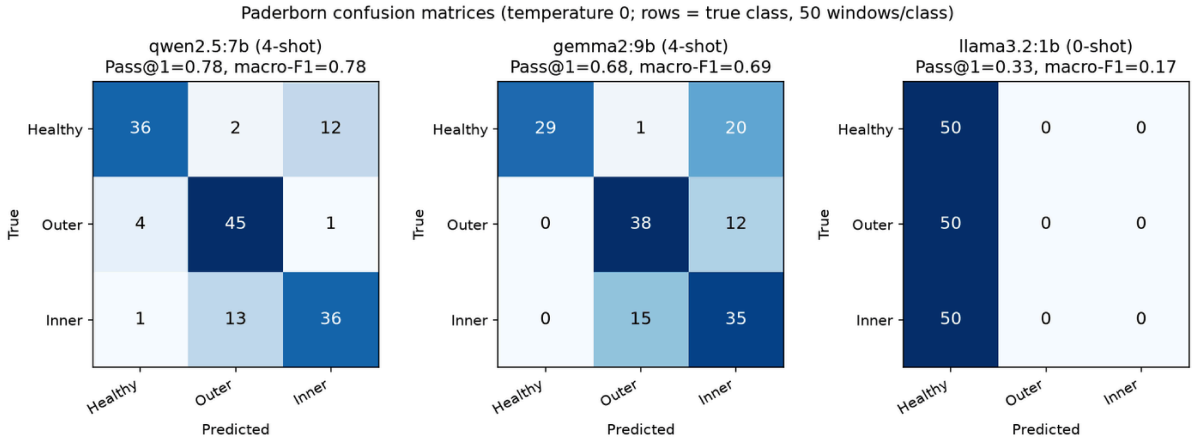

Figure 3. Representative Paderborn confusion matrices (temperature 0; rows = true class, 50 windows per class): two capable models versus the collapsed llama3.2:1b, which labels every window "Healthy".

3.3. Finding 1: A Model-Capability Gate

The clearest pattern in the results is that model size, not prompt design, decides whether a model can perform the task at all. Models at roughly 7B parameters and above rise consistently above the 0.33 majority-class floor (Figure 4); model capability, not prompt engineering, is the binding constraint. The ~7B models reach a peak Pass@1 of roughly 0.73 to 0.78 on Paderborn, while the 1B and 2B models never move meaningfully off the 0.33 floor no matter how many examples they are shown. The strongest of the smaller models is qwen2.5:3b, which climbs to about 0.62 at its best; it shows that training quality can partly offset size, but it stays below the 7B tier and is less steady.

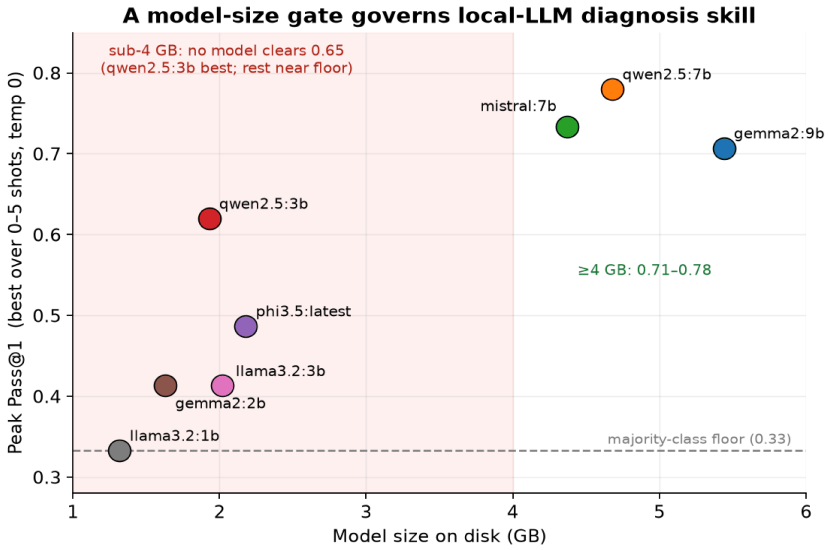

Figure 4. Peak Pass@1 versus model size on disk.

3.4. Finding 2: Few-Shot Is Non-Monotonic

More context does not translate into better accuracy in a simple, monotonic way. Accuracy typically rises to a 4-shot sweet spot, then declines (Figure 5); phi3.5 breaks down entirely at ≥4-shot on all three datasets. For the two strongest 7B models, the peak sits at four examples per class (qwen2.5:7b reaches 0.78 and mistral:7b 0.73 on Paderborn) before longer prompts begin to hurt, while gemma2:9b keeps improving slightly out to five shots. The phi3.5 failure is the most striking: its valid-output rate falls to about 0.13 at four and five shots, so it stops returning usable answers rather than merely losing accuracy.

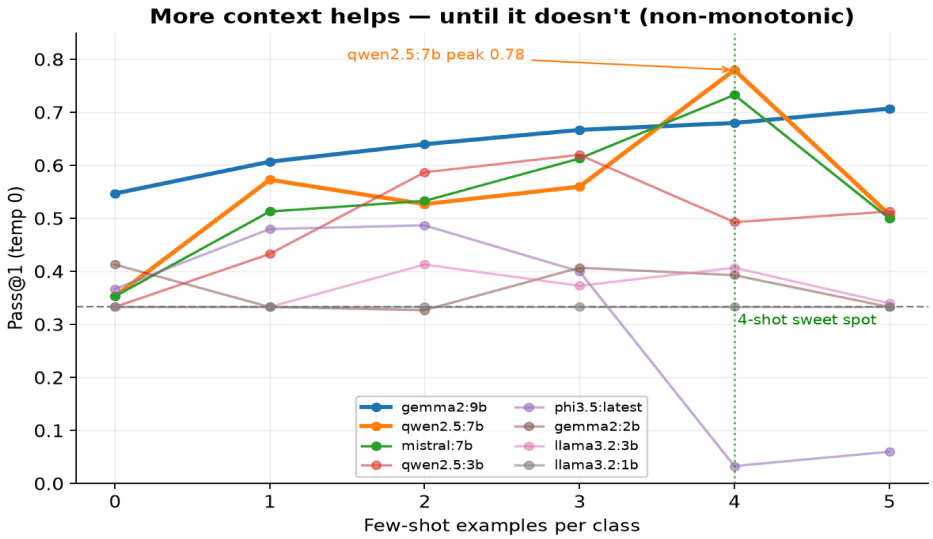

Figure 5. Pass@1 versus number of few-shot examples per class.

3.5. Finding 3: Accuracy Is Not Reliability

A model that is right on average is not the same as a model that can be trusted on any single run. At temperature 0.7, self-consistency separates steady from volatile models (Figure 6). llama3.2:1b appears "consistent" only because it always predicts the same wrong class. Gemma2:9b is both accurate and steady, agreeing with itself on roughly 87 to 97 percent of windows across repeated runs, while some mid-sized models swing from run to run even when their average accuracy looks acceptable. This is why we treat reliability as a second axis and, in Section 3.8, turn agreement across models into an explicit trust signal.

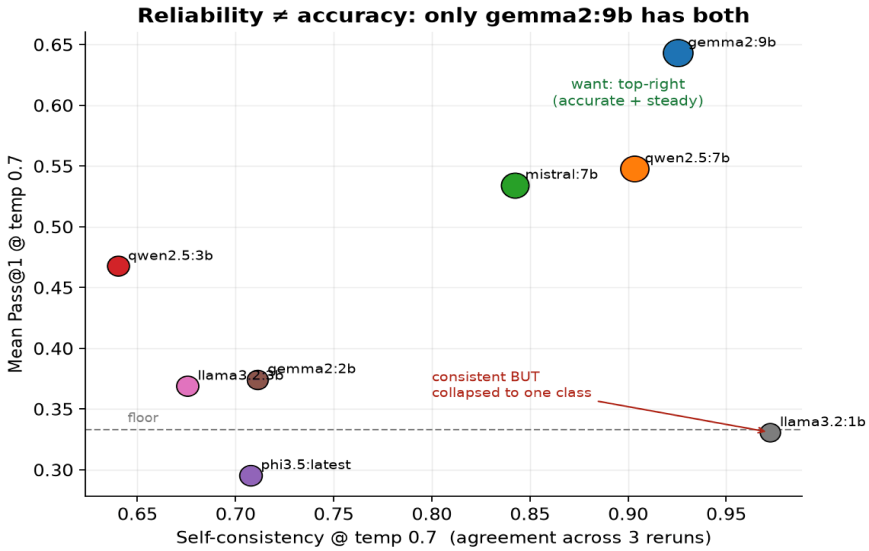

Figure 6. Accuracy versus self-consistency (temperature 0.7).

3.6. Finding 4: One Model Leads on Both Axes, on All Three Datasets

Taken together, the accuracy and reliability axes point to the same model. A single model, gemma2:9b, is the most accurate real model on all three datasets and among the steadiest (smallest or near-smallest Pass@1 → Pass all 3 gap); on HUST, qwen2.5:7b has a marginally smaller gap (Table 3). Its mean Pass@1 leads on every dataset (about 0.64 on Paderborn, 0.75 on CWRU, and 0.60 on HUST), and its drop from Pass@1 to the stricter Pass all 3 metric is small, which is what makes it dependable rather than merely good on average. That the same 5.4 GB model wins on both axes across three independent test rigs is the result that matters for an SME, because it removes the need to pick a different model for each machine.

Table 3. Mean Pass@1 and Pass@1 → Pass all 3 gap (temperature 0.7). A smaller gap indicates a steadier model; the best Pass@1 per dataset is in bold.

	Paderborn		**CWRU**		**HUST**	
Model	**Pass@1**	**Gap**	**Pass@1**	**Gap**	**Pass@1**	**Gap**
gemma2:9b	**0.643**	0.036	**0.749**	0.020	**0.604**	0.054
qwen2.5:7b	0.548	0.039	0.629	0.021	0.500	0.038
mistral:7b	0.535	0.061	0.699	0.039	0.549	0.090
qwen2.5:3b	0.468	0.155	0.516	0.116	0.539	0.192
llama3.2:1b (collapsed)	0.331	0.001	0.334	0.002	0.285	0.004

3.7. Cross-Dataset Replication

The strongest evidence that these findings are real, and not an artifact of one test rig, is that they reappear on two further datasets collected by different groups on different hardware. All four findings hold on the second and third datasets (Figure 7): the model ordering is preserved, gemma2:9b is the most accurate real model on all three, phi3.5 collapses at high shots on all three (on HUST its valid-output rate falls below 0.15 at 4–5-shot), and the "consistent but collapsed" trap recurs (on HUST llama3.2:1b predicts essentially one class: self-consistency 0.99 but Pass@1 0.28). CWRU scores are uniformly higher (cleaner dataset), and HUST is the hardest of the three, but the patterns, not the absolute numbers, are what transfer. CWRU confusion matrices are shown in Figure 8. The absolute accuracies shift with dataset difficulty, but the ordering of the models and the shape of every finding are preserved. For a practitioner, this is the reassuring part: the

same model and the same prompt carry from one machine to another without re-tuning. The absolute accuracies differ across the three datasets because they differ in sensor and mounting, sampling rate (64 kHz for Paderborn, 12 kHz for CWRU, and 25.6 kHz for HUST), bearing geometry, and operating condition, all of which change the envelope-spectrum features. What replicates is the ordering of the models and the shape of each finding, which is dataset-level replication rather than machine-identity-level or field-condition generalization; we do not claim the latter.

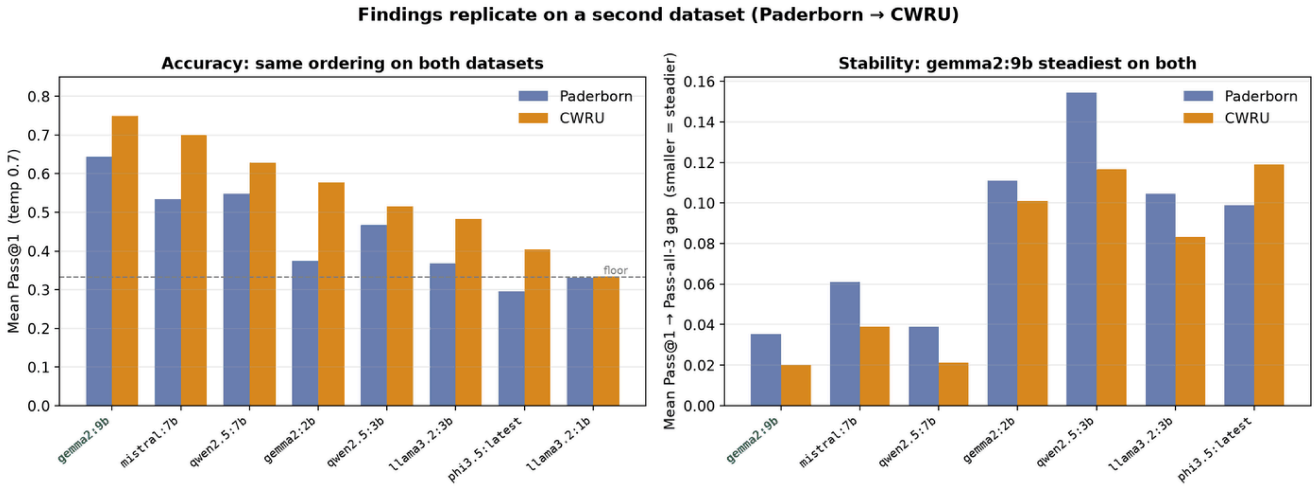

Figure 7. Findings replicate on a second dataset (Paderborn → CWRU).

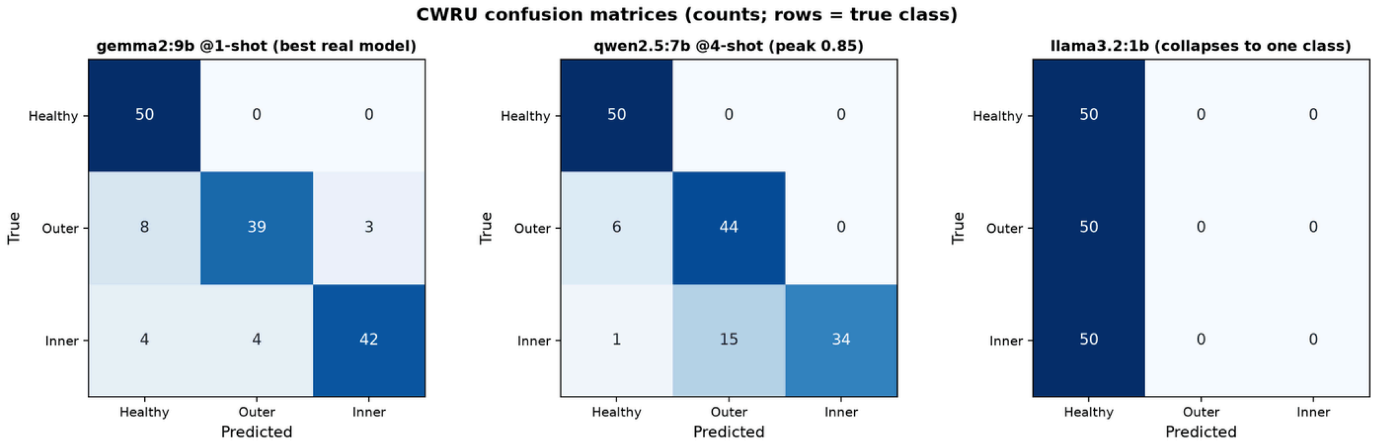

Figure 8. Representative CWRU confusion matrices (counts; rows are the true class).

3.8. Ensemble Agreement Gating: A Training-Free Reliability Method

Table 4 and Figure 9 evaluate the gating rule (Section 2.5) at 4-shot on all three datasets. Majority voting alone does not beat the best single model (Paderborn 0.767 vote vs. 0.767 best single; CWRU 0.827 vs. 0.853), so we claim no accuracy gain from voting. The *agreement level*, however, is a strong trust signal: keeping only windows where all three models agree raises accuracy on the answered subset on every dataset: Paderborn 0.77→0.87 (60% coverage), CWRU 0.83→0.86 (59%), and HUST 0.56→0.70 (21%). An operator can therefore auto-accept the high-agreement majority and flag the remainder for inspection, using only free local models and no labels, directly addressing the "which predictions can I trust?" question an SME faces in deployment.

Table 4. Ensemble agreement gating at 4-shot (temperature 0.7) over three models (gemma2:9b, qwen2.5:7b, mistral:7b). "Gated" keeps only windows where all three models agree.

Dataset	Best Single	Majority Vote	Gated Acc. (3/3)	Coverage
Paderborn	0.767	0.767	**0.867**	0.60
CWRU	0.853	0.827	**0.864**	0.59
HUST	0.664	0.564	**0.700**	0.21

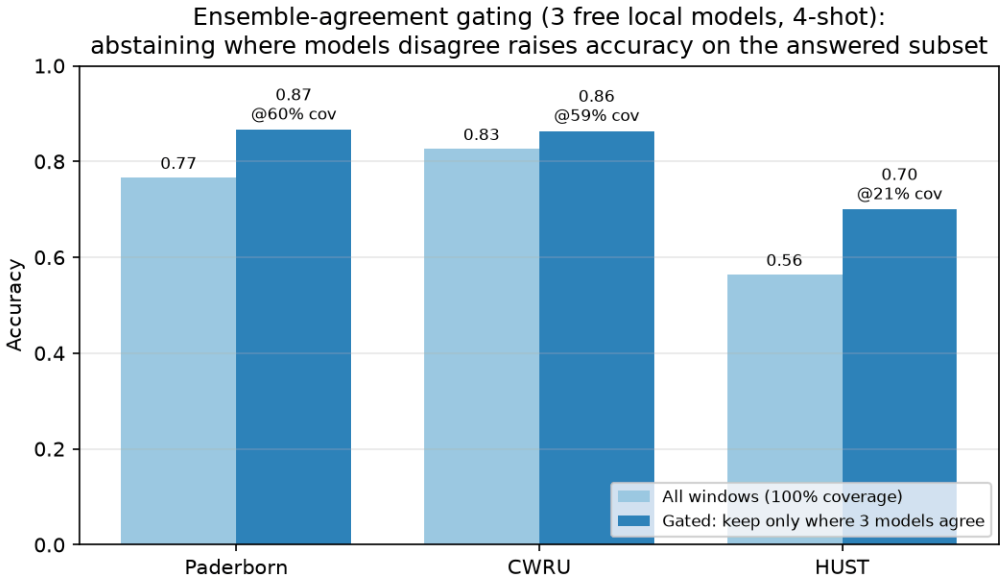

Figure 9. Ensemble agreement gating: abstaining on windows where three free models disagree raises accuracy on the answered subset across all three datasets (coverage annotated).

To characterize the full selective-prediction trade-off rather than a single operating point, Table 4 should be read together with the following risk–coverage behavior. As the agreement threshold increases, coverage falls but accuracy on the retained windows rises and the selective risk falls: on Paderborn, accuracy is 0.72 at full coverage, 0.78 when at least two of three models agree (0.91 coverage), and 0.87 at unanimous agreement (0.60 coverage); on CWRU, 0.81, 0.82, and 0.86 (coverage 1.00, 0.99, 0.59); and on HUST, 0.56, 0.59, and 0.70 (coverage 1.00, 0.91, 0.21). The rejected windows are markedly less accurate than the retained ones; for example, 0.14 on the Paderborn windows where all three models disagree, which is the behavior a useful gate should show. The HUST unanimous point reaches 0.70 accuracy only at 0.21 coverage, meaning the rule abstains on about four windows in five; we regard this as a marginal operating point rather than a practical gain, whereas Paderborn and CWRU retain about 60 percent of windows with a clear accuracy improvement. Cross-model agreement therefore predicts correctness. Single-model self-consistency, by contrast, does not on its own: across all model-and-shot cells its correlation with accuracy is near zero, and a collapsed model such as llama3.2:1b is almost perfectly self-consistent while sitting at the majority-class floor, which is why the gate combines several different models rather than repeated runs of one model. Self-consistency is defined as the fraction of test windows on which all three temperature 0.7 reruns return the same label, and the headline numbers are reported with bootstrap 95% confidence intervals. We also note a limitation of this gating analysis: the three models used for the ensemble (gemma2:9b, qwen2.5:7b, and mistral:7b) are the three most accurate models in Table 3, so the agreement signal is measured among already-strong models. Agreement among models selected for accuracy is not by itself evidence that inter-model agreement is a universal trust signal, and a fully model-agnostic ensemble would be a stronger test of that claim.

3.9. Head-to-Head Against a Frontier Model

To test whether paying for a frontier model buys better diagnosis, we ran a current frontier model (Claude Sonnet) through the identical prompt, features, test windows, and grading. Any difference is therefore model quality, not prompting. The free local models match it in the settings we evaluated (Table 5, Figure 10).

Table 5. Frontier versus best free/local model, identical prompt and data (Pass@1). The HUST row is the 0-shot comparison.

Setting	Frontier (Claude Sonnet)	Best Free/Local
Paderborn 0-shot	0.540	0.540 (gemma2:9b)
Paderborn 4-shot	0.760	**0.767** (qwen2.5:7b)
CWRU 0-shot	0.493	**0.553** (gemma2:9b)
CWRU 4-shot	0.827	**0.853** (qwen2.5:7b)
HUST 0-shot	0.479	**0.521** (gemma2:9b)

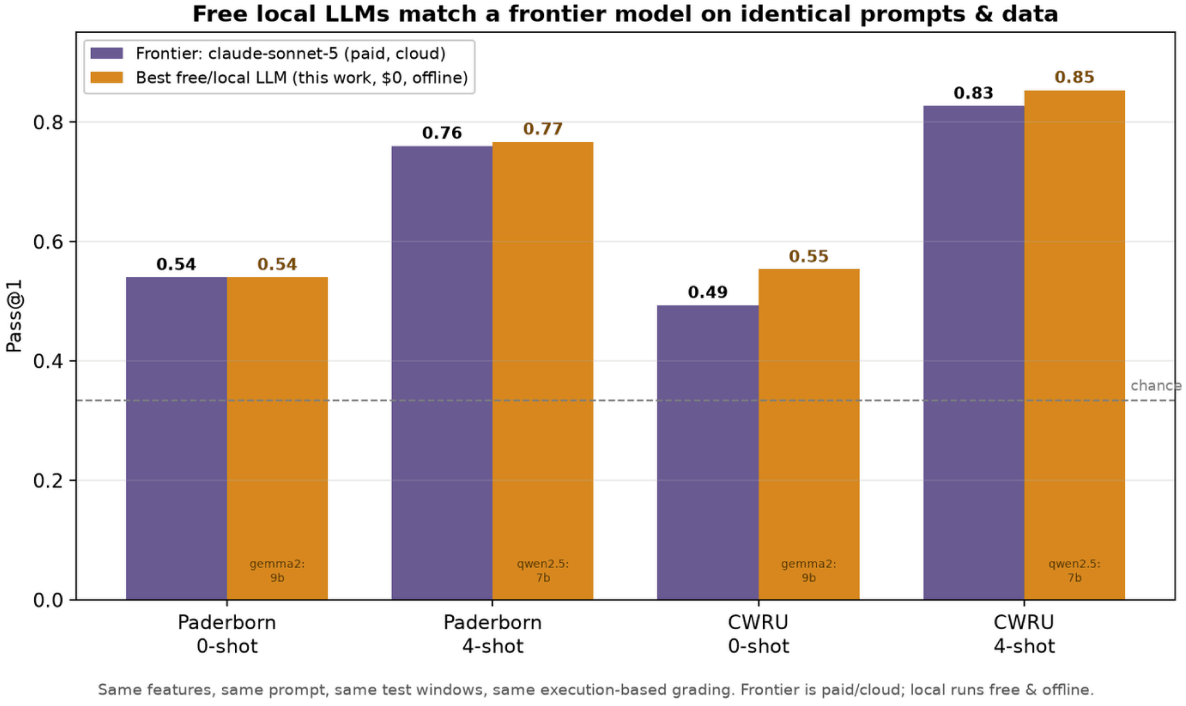

Figure 10. A paid frontier model confers no accuracy advantage over free, local models on this task.

For reproducibility, the frontier model is claude-sonnet-5, accessed on 1 August 2026, queried at temperature 0 with a 512-token limit and a single response per window, using the identical system prompt, feature text, test windows, and grading as the local models. The comparison covers only the settings listed in Table 5 (Paderborn and CWRU at 0 and 4 shots, HUST at 0 shots); we restrict the claim to these settings and note that the closest values, frontier 0.76 against local 0.767 on Paderborn at four shots, lie within the bootstrap confidence interval and are therefore not distinguishable. The frontier comparison is a targeted control rather than an exhaustive grid: its only purpose is to test whether a paid model buys an accuracy advantage. The zero-shot setting, which is the label-free headline of this study, is reported for all three datasets, and the four-shot cell is added for the two datasets where we probe the strongest local operating point. On HUST, the best free local model already matches or exceeds the frontier at zero-shot (0.52 versus 0.48), so the no-advantage conclusion holds for HUST without the four-shot cell; across every

evaluated setting, the frontier never exceeds the best local model, and the claim is limited to those settings.

3.10. What the Training-Free Approach Replaces

The value of an off-the-shelf, training-free method is clearest against the alternatives an SME would otherwise face (Table 6). **Supervised pipelines** reach near-ceiling in-distribution accuracy on the identical features (random forest 0.998 Paderborn/0.958 CWRU; our prior raw-signal CNN 0.993), but only after a per-machine cycle of data collection, labeling, training, and validation by ML staff, and they do not transfer; the CNN collapses to 0.29–0.40 on unseen rigs (Figure 11). **Fine-tuned LLMs** [5] reach ~96.5% but require LoRA/QLoRA fine-tuning on labeled target data (ML expertise and compute per setup). **Frontier LLM agents** [4] (best ~68% task completion with a paid cloud model) require a paid, cloud-hosted model. The off-the-shelf free/local approach requires none of these: no labeled data, no training, no ML staff, no cloud, and the same model and prompt work unchanged across all three datasets.

Table 6. What each approach demands of an SME.

Approach	Labeled Data	Per-Machine Training	ML Expertise	Cloud	Cross-Dataset
Supervised classifier	required	required	required	no	retrain each time
Frontier LLM agent	none	none	low	**required**	yes
Free/local LLM (this work)	**none**	**none**	**none**	**no**	**yes, unchanged**

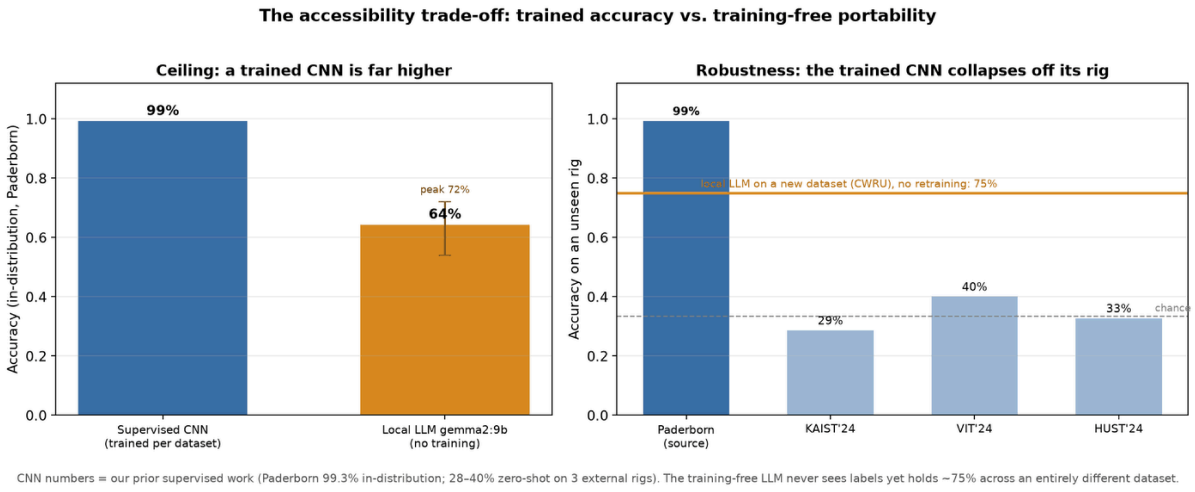

Figure 11. Trained accuracy versus training-free portability: the supervised CNN is accurate in distribution but collapses off its rig.

3.11. Comparison with Simple Baselines on the Same Features

A central question, raised by the reviewers, is whether the diagnostic signal comes from the language model's reasoning or from the engineered features themselves. To separate the two, we trained standard classifiers on the same feature text and evaluated them on the identical test windows used for the LLMs, with the trained models fitted only on the disjoint reference split. Table 7 reports the results. A logistic regression, a shallow (depth-2) decision tree, and a random forest all substantially outperform every local LLM, and even an untrained rule based only on the BPFO-to-BPFI energy ratio exceeds the best local model on two of the three datasets. Using its strongest few-shot setting, the best local LLM reaches 0.78, 0.87, and 0.70, which still sit below the trained baselines on all three datasets; the gaps exceed the bootstrap 95% confidence intervals. This shows that

on these separable hand-crafted features the discriminative work is done largely by the features, and that a simple supervised model is the better choice when the features and a few labels are available. We therefore present this study as a benchmark and cautionary result rather than a claim that local LLMs are competitive classifiers; the value of the free local approach lies in requiring no machine-learning pipeline and in the reliability gating of Section 3.8, not in accuracy. We also note that computing the envelope spectrum and the characteristic-defect-frequency features itself requires signal-processing expertise, which every method compared here shares, so the free local route removes the machine-learning pipeline rather than all expertise.

Table 7. Simple baselines versus the local LLMs on the identical feature text and test windows (accuracy). Trained baselines are fitted on the labeled reference split; the physics rule and the LLMs use no labels. Bootstrap 95% confidence intervals are reported in the text.

Method	Paderborn	CWRU	HUST	Labels
Logistic regression	0.97	1.00	0.99	yes
Decision tree (depth 2)	0.83	0.91	0.96	yes
Random forest	1.00	0.97	0.99	yes
Physics BPFO/BPFI rule	0.74	0.81	0.46	no
Best local LLM (few-shot)	0.78	0.87	0.70	no
Best local LLM (zero-shot)	0.55	0.55	0.52	no

4. Discussion

For an SME that cannot label data, retrain per machine, or hire ML staff, a single 5.4 GB general-purpose model (gemma2:9b) delivers roughly 0.7–0.87 accuracy with quantified reliability, offline and free, and the identical setup works on two further datasets with no changes. This is the core claim: useful bearing diagnosis without a per-setup AI project. A supervised classifier remains superior *when abundant labeled data and ML expertise are available* exactly the resources an SME lacks. The size gate gives a deployment guideline (target $\geq$7B parameters), the few-shot ablation a sensible prompt length (~2–4 examples per class), and the reliability axis warns against trusting small "self-consistent" models. That a paid frontier model confers no accuracy advantage on this feature-based task removes the last practical reason to incur cloud cost or send data off-site. We now qualify this: on the same features, a simple supervised baseline outperforms the local models (Section 3.11), so the free local approach is attractive only where labeled data and machine-learning staff are genuinely unavailable, and its contribution is the training-free setup and the reliability gating rather than accuracy.

Construct validity. The models classify hand-crafted features rather than the raw signal, so the results reflect the feature set as much as the model; execution-based exact-match grading also credits only the final label, not the (sometimes partially correct) justification. *Internal validity.* The LLM evaluation uses a single seed and 50 windows per class, so absolute numbers carry sampling noise; reliability is estimated from three reruns at temperature 0.7. Splits are made at the recording level; the zero-shot setting is leak-free by construction, and few-shot exemplars are drawn from disjoint recordings, but we do not claim bearing-identity-level generalization, a known confound in supervised bearing studies. *External validity.* We cover three public datasets, a three-class task (Ball faults excluded for label parity), and laboratory test-rig data rather than field deployments; both the local models and the frontier baseline (Claude Sonnet) are specific versioned checkpoints, so exact numbers may drift as models are updated. *Reproducibility.* Every configuration is autosaved, and the code is public, but the frontier comparison depends on a hosted model that may change over time.

5. Conclusions

This study set out to determine whether general-purpose, openly available LLMs that run locally on commodity hardware can diagnose bearing faults from vibration features without any task-specific training, and how far such a solution can be trusted. Across three standard datasets, Paderborn, CWRU, and HUST, evaluated under a single fixed, leak-free protocol, the answer is that small, free, local LLMs can indeed diagnose bearing faults, though less accurately than a simple supervised baseline on the same features. A single 5.4 GB general-purpose model, gemma2:9b, emerged as a clear best choice: it delivered roughly 0.70–0.87 accuracy with quantified reliability, offline and at no cost, and the identical model and prompt worked on all three datasets with no changes. This supports the core claim of the paper, that useful bearing diagnosis is achievable without a per-machine labeling, training, or fine-tuning project. As Section 3.11 shows, a simple supervised classifier on the same features is more accurate, so the contribution of this work is the honest cross-dataset benchmark and the training-free reliability gating rather than a claim that local LLMs are competitive classifiers.

Beyond the headline result, the benchmark yields concrete deployment guidance. A model-size gate indicates that capability tended to rise around the 7B mark for the models we tested, so practitioners should consider models of at least this size, while noting that architecture and training also matter; a few-shot ablation shows that a small number of examples per class, roughly two to four, is sufficient and that adding more is not reliably beneficial; and a reliability axis based on self-consistency warns against trusting small models that merely appear confident. We further found that a current paid frontier model confers no accuracy advantage on this feature-based task, which removes the last practical reason for an SME to incur cloud cost or send data off-site. The paper's methodological contribution is ensemble agreement gating: a training-free rule that combines several free, local models by majority vote and abstains when they disagree, turning the reliability axis into a deployable selective-prediction mechanism. It flags low-confidence cases for human review and raises accuracy on the predictions it retains, at no additional training or labeling cost. These results are naturally bounded by the study's scope. The models classify hand-crafted features rather than raw signals; the evaluation uses laboratory test-rig data and a three-class task rather than field deployments; and both the local models and the frontier baseline are specific versioned checkpoints, so exact numbers may shift as models are updated. These considerations qualify the magnitude of the reported numbers rather than the central finding, which holds consistently across three independent datasets. The most promising direction for future work is an agentic variant: giving the capable 7B-class models actual tools, namely feature extraction and envelope-spectrum analysis, inside a reasoning loop, producing a true local counterpart to agentic PHM benchmarking. Such a system would be informative under either outcome and would extend the present training-free, offline approach from feature classification toward more autonomous diagnosis.

Author Contributions: Conceptualization, M.H.S. and P.K.; methodology, M.H.S.; software, M.H.S.; validation, M.H.S.; formal analysis, M.H.S.; investigation, M.H.S.; data curation, M.H.S.; writing—original draft preparation, M.H.S.; writing—review and editing, P.K.; visualization, M.H.S.; supervision, P.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Woosong University Academic Research 2026.

Data Availability Statement: The three bearing datasets analyzed in this study are publicly available from their original providers: the Paderborn University bearing dataset, the Case Western Reserve University (CWRU) bearing dataset, and the HUST bearing dataset. No new data were created in this study. The code used to generate the results is available from the corresponding author upon reasonable request as it is part of an ongoing research project.

Acknowledgments: During the preparation of this manuscript, the authors used generative AI-based tools to assist with editing and refining the language of the text and with preparing figures from the authors' own experimental data. All scientific content, analysis, and conclusions are the authors' own. The authors reviewed and verified all output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BPFI	Ball pass frequency, inner race
BPFO	Ball pass frequency, outer race
BSF	Ball spin frequency
CNN	Convolutional neural network
CWRU	Case Western Reserve University (bearing dataset)
FTF	Fundamental train frequency
HUST	Huazhong University of Science and Technology (bearing dataset)
LLM	Large language model
ML	Machine learning
PHM	Prognostics and health management
RMS	Root mean square
SME	Small or medium enterprise
SVM	Support vector machine

Appendix A. Feature Extraction Pipeline and Example

Every one-second window is processed identically for all three datasets. Time-domain features are RMS = sqrt(mean(x^2)), peak-to-peak = max(x) − min(x), crest factor = peak/RMS, shape factor = RMS/mean(|x|), impulse factor = peak/mean(|x|), clearance factor = peak/(mean(sqrt(|x|)))^2, and kurtosis and skewness (the fourth and third standardized moments of x). The envelope spectrum is obtained by band-pass filtering with a fourth-order Butterworth filter between 1 and 5 kHz, taking the magnitude of the Hilbert analytic signal, removing its mean, and applying a real FFT; the amplitude at a characteristic frequency is the peak within a plus or minus 3 Hz band around that frequency and its first three harmonics, with bpfo_energy and bpfi_energy summing the first three BPFO and BPFI harmonics and bpfo_to_bpfi_ratio their ratio. The characteristic frequencies are BPFO = (n/2)(1 − (d/D)cos f) fr, BPFI = (n/2)(1 + (d/D)cos f) fr, FTF = (1/2)(1 − (d/D)cos f) fr, and BSF = (D/2d)(1 − ((d/D)cos f)^2) fr, with n balls, ball diameter d, pitch diameter D, contact angle f, and shaft frequency fr (Paderborn 6203, n = 8, d = 6.75 mm, D = 28.5 mm, f = 0; CWRU 6205, n = 9; HUST ER-16K, n = 9). The 21 features are supplied in this order, to three decimal places: rms, peak_to_peak, crest_factor, kurtosis, skewness, impulse_factor, clearance_factor, spectral_centroid_hz, spectral_entropy, shaft_freq_hz, bpfo_hz, bpfi_hz, env_bpfo_1x, env_bpfo_2x, env_bpfo_3x, bpfo_energy, env_bpfi_1x, env_bpfi_2x, env_bpfi_3x, bpfi_energy, bpfo_to_bpfi_ratio. Ollama model tags can change over time, so the exact tags and pull dates are recorded with the released code. One complete example of the feature text for an outer-race window (each feature is on its own line in the prompt, shown here separated by semicolons): rms: 0.482; peak_to_peak: 5.112; crest_factor: 6.058; kurtosis: 5.414; skewness: −0.456; impulse_factor: 8.212; clearance_factor: 10.060; spectral_centroid_hz: 8317.410; spectral_entropy: 0.917; shaft_freq_hz: 24.997; bpfo_hz: 76.306; bpfi_hz: 123.668; env_bpfo_1x: 8184.375; env_bpfo_2x: 4202.570; env_bpfo_3x: 1985.647; bpfo_energy: 14372.591; env_bpfi_1x: 917.545; env_bpfi_2x: 986.856; env_bpfi_3x: 1129.068; bpfi_energy: 3033.470; bpfo_to_bpfi_ratio: 4.738.

References

1. Kumar, P.; Hati, A.S. Review on Machine Learning Algorithm Based Fault Detection in Induction Motors. *Arch. Comput. Methods Eng.* **2021**, *28*, 1929–1940. [CrossRef]
2. Rohan, A.I.; Ridita, T.A.; Anonto, H.Z.; Hossain, M.I.; Shufian, A.; Mahin, M.S.R.; Islam, M.A. Intelligent Fault Diagnosis in Rolling Element Bearings: Combining Envelope Spectrum and Spectral Kurtosis for Enhanced Detection. *Results Eng.* **2025**, *27*, 106899. [CrossRef]
3. Neupane, D.; Seok, J. Bearing Fault Detection and Diagnosis Using Case Western Reserve University Dataset With Deep Learning Approaches: A Review. *IEEE Access* **2020**, *8*, 93155–93178. [CrossRef]
4. Das, A.; Patel, D. PHMForge: A Scenario-Driven Agentic Benchmark for Industrial Asset Lifecycle Maintenance. *arXiv* **2026**, arXiv:2604.01532.
5. Huang, K.; Chen, Q. Motor Fault Diagnosis and Predictive Maintenance Based on a Fine-Tuned Qwen2.5-7B Model. *Processes* **2025**, *13*, 2051. [CrossRef]
6. Li, Y.-F.; Wang, H.; Sun, M. ChatGPT-like Large-Scale Foundation Models for Prognostics and Health Management: A Survey and Roadmaps. *Reliab. Eng. Syst. Saf.* **2024**, *243*, 109850. [CrossRef]
7. Bao, P.; Yi, W.; Zhu, Y.; Shen, Y.; Peng, H. A Spectral Interpretable Bearing Fault Diagnosis Framework Powered by Large Language Models. *Sensors* **2025**, *25*, 3822. [CrossRef] [PubMed]
8. Peng, H.; Gao, J.; Liu, J.; Du, J.; Wang, W. A Unified Rotating Machinery Health Management Framework Leveraging Large Language Models for Diverse Components, Conditions, and Tasks. *Eng. Appl. Artif. Intell.* **2025**, *162*, 112544. [CrossRef]
9. Di Maggio, L.G. Toward Autonomous LLM-Based AI Agents for Predictive Maintenance: State of the Art, Challenges, and Future Perspectives. *Appl. Sci.* **2025**, *15*, 11515. [CrossRef]
10. Lessmeier, C.; Kimotho, J.K.; Zimmer, D.; Sextro, W. 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. In Proceedings of the PHM Society European Conference, Bilbao, Spain, 5–8 July 2016; Volume 3.
11. Smith, W.A.; Randall, R.B. Rolling Element Bearing Diagnostics Using the Case Western Reserve University Data: A Benchmark Study. *Mech. Syst. Signal Process.* **2015**, *64–65*, 100–131. [CrossRef]
12. Thuan, N.D.; Hong, H.S. HUST Bearing: A Practical Dataset for Ball Bearing Fault Diagnosis. *BMC Res. Notes* **2023**, *16*, 138. [CrossRef] [PubMed]
13. Chen, M.; Tworek, J.; Jun, H. Evaluating Large Language Models Trained on Code. *arXiv* **2021**, arXiv:2107.03374.
14. Liu, Y.; Shen, B.; Wong, D.S.-H.; Jia, M.; Yao, Y. Explainable Neural Network Meets Graph Neural Network: Recent Advances in Process Fault Detection and Diagnosis. *Comput. Chem. Eng.* **2026**, *206*, 109528. [CrossRef]
15. Yang, Z.; Mao, L.; Ye, L.; Ma, Y.; Song, Z.; Chen, Z. AKGNN: When Adaptive Graph Neural Network Meets Kolmogorov–Arnold Network for Industrial Soft Sensors. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 1–13. [CrossRef]
16. Yang, Z.; Chen, J.; Bi, Z.; Jiang, X.; Yao, L.; Lou, J.; Song, Z. CLARI: Constrained Reflective LLM Inference for Few-Shot Soft Sensing under Operating-Condition Drift. *Control Eng. Pract.* **2026**, *177*, 107187. [CrossRef]
17. Pérez-Pérez, E.-J.; López-Estrada, F.-R.; Puig, V.; Valencia-Palomo, G.; Santos-Ruiz, I. Fault Diagnosis in Wind Turbines Based on ANFIS and Takagi–Sugeno Interval Observers. *Expert Syst. Appl.* **2022**, *206*, 117698. [CrossRef]
18. Pérez-Pérez, E.-J.; Puig, V.; Santos-Ruiz, I.; Gúzman-Rabasa, J.-A.; Valencia-Palomo, G. Wind Turbine Fault Diagnosis Using Structural Analysis and Optimized-Rectangular GPR Interval Estimation. *Control Eng. Pract.* **2026**, *172*, 106940. [CrossRef]
19. Meta Llama Team. The Llama 3 Herd of Models. *arXiv* **2024**, arXiv:2407.21783.
20. Gemma Team. Gemma 2: Improving Open Language Models at a Practical Size. *arXiv* **2024**, arXiv:2408.00118.
21. Qwen Team. Qwen2.5 Technical Report. *arXiv* **2024**, arXiv:2412.15115.
22. Abdin, M.; Aneja, J.; Awadalla, H.; Awadallah, A.; Awan, A.A.; Bach, N.; Bahree, A.; Bakhtiari, A.; Bao, J.; Behl, H.; et al. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. *arXiv* **2024**, arXiv:2404.14219.
23. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. *arXiv* **2023**, arXiv:2310.06825.
24. Ollama: Get Up and Running with Large Language Models Locally. Available online: https://ollama.com (accessed on 10 August 2026).