Leakage-Safe, Reproducible Benchmarking for Vibration-Based Fault Diagnosis

Knap Pawel[1], Jachymczyk Urszula[1], and Lalik Krzysztof[1]

[1] *AGH University of Krakow, Kraków, 30-059, Poland*
pknap@agh.edu.pl
ujachymczyk@agh.edu.pl
klalik@agh.edu.pl

ABSTRACT

Vibration-based bearing fault diagnosis is a widely studied predictive maintenance problem, but reported results are often difficult to compare. Performance depends not only on the model itself, but also on the evaluation protocol, the train–test split, and the type of domain shift considered. In particular, leakage-prone window-level splitting and loosely defined source–target settings can lead to overly optimistic conclusions that do not reflect real transfer performance across changing operating conditions, acquisition regimes, or bearing identities. To address this issue, this paper introduces a leakage-safe and reproducible benchmark for cross-domain bearing fault diagnosis on the Case Western Reserve University and Paderborn University datasets. The benchmark defines six fixed source–target scenarios, enforces recording-level train–test separation, and evaluates both machine-learning and deep-learning baselines under a common protocol. Final reporting is based on a consistent evaluation setup, with repeated-seed follow-up used where necessary to support reliable conclusions for deep-learning models. The results show that scenario difficulty is highly heterogeneous. Some transfer settings are effectively saturated, while others remain substantially more challenging. Deep-learning models often achieve stronger performance, but their conclusions can be sensitive to initialization and require repeated-seed validation. Overall, the benchmark provides a reproducible basis for scenario-level evaluation and more reliable comparison of cross-domain bearing diagnosis methods. The code for this study is publicly available at `https://github.com/1Sensor/pdm-bench`.

1. INTRODUCTION

Vibration-based predictive maintenance (PdM) is one of the most widely studied application areas for machine learning (ML) and deep learning (DL) in rotating machinery diagnostics. PdM covers a broad range of tasks, from fault detection and diagnosis to prognosis, remaining useful life estimation, and maintenance decision support. The present study focuses on one specific part of this broader framework: vibration-based bearing fault diagnosis (FD). Rolling element bearings are critical components in industrial assets, and their degradation is frequently monitored through vibration analysis because bearing faults can lead to unplanned downtime, reduced efficiency, and costly maintenance interventions (Lei et al., 2020; Smith & Randall, 2015a).

A large body of work reports increasingly strong classification results on public bearing datasets, yet meaningful comparison across studies remains difficult (Apicella, Isgrò, & Prevete, 2025; Rosa, Braga, & Silva, 2024; Vieira, Bauler, Rosa, & Silva, 2026). In many cases, reported gains depend not only on the model itself, but also on preprocessing choices, window extraction procedures, and, critically, the train–test partitioning protocol, which may introduce data leakage (Matania, Cohen, Bechhoefer, & Bortman, 2024). When these design choices are not controlled or are only partially documented, comparison across studies becomes difficult and the literature can accumulate apparently strong results that are not directly comparable. As a result, it is often unclear whether an observed improvement reflects a genuinely better method or simply a more favorable evaluation protocol.

Signals acquired for PdM systems are rarely random and tend to have a sequential character. With a random splitting strategy, the diagnostic algorithm may learn patterns that occur in the future relative to the test point. When windows derived from the same recording, run, or bearing instance are allowed to appear in both training and test sets, the evaluation may become overly optimistic (Matania et al., 2024; Neupane,

Bouadjenek, Dazeley, & Aryal, 2025; Zhao et al., 2020). Under such conditions, a model may exploit acquisition-specific or bearing-specific signatures instead of learning fault-related structure that transfers to unseen data (Apicella et al., 2025; Vieira et al., 2026). The issue is particularly relevant when recordings are segmented into overlapping windows and those windows are subsequently assigned to the training and test sets. In such cases, the model may exploit low-level correlations tied to a specific acquisition rather than learn patterns that transfer across operating conditions, fault instances, or bearing identities. A benchmark that ignores this issue can produce impressive numbers while saying little about real generalization. This may result in a significant drop in PdM system efficiency as operating conditions change. Therefore, it is crucial to evaluate model generalization under changing conditions. This is also one of the main reasons for the strong interest in domain adaptation techniques for bearing fault diagnosis (Chen et al., 2023).

To obtain measurable and trustworthy model evaluation, there has been growing interest in more rigorously defined evaluation protocols and benchmark studies. Even condition-wise splits can remain problematic because the same faulty bearings may be measured under multiple operating conditions, allowing bearing identity to leak across partitions (Hendriks, Dumond, & Knox, 2022; Abburi et al., 2023; Rosa et al., 2024). Recent benchmark-oriented studies have shown that a multi-label reformulation combined with more realistic split design can mitigate several important limitations of conventional evaluation and support the use of receiver operating characteristic (ROC) analysis and the area under the ROC curve (AUROC) rather than relying solely on accuracy (Rosa et al., 2024). Other recent work has further emphasized the importance of leakage-free, bearing-wise evaluation across datasets such as Case Western Reserve University (CWRU) bearing dataset, Paderborn University (PU) bearing dataset, and the University of Ottawa Rolling Element Bearing Dataset–Vibration and Acoustic Fault Classification under Load and Speed conditions (UORED-VAFCLS), demonstrating that performance can change substantially when partitioning is made physically meaningful (Vieira et al., 2026).

1.1. Research Gap

Although evaluation protocols in vibration-based FD have received increasing attention, they remain insufficiently standardized. Studies often rely on different subsets of the same datasets, preprocessing pipelines, feature representations, and data-splitting strategies. Consequently, even when identical benchmark datasets are used, the reported results are often not directly comparable (Rosa et al., 2024; Vieira et al., 2026). This creates a scientific challenge: a large number of isolated results tied to specific protocol settings, but much more limited reliable and reproducible evidence that clearly demonstrates the advantage of one algorithm over another.

Although interest in more cohesive evaluation protocols, stricter experimental standards, leakage-safe assessment, and benchmarking has grown, there is still no benchmark that simultaneously provides a comparison of ML and DL models, cross-domain scenarios for testing model generalization under changing conditions, and an analysis of the trade-off between computational cost and result quality.

1.2. Contribution

In this work, we address this gap by introducing a unified, leakage-safe, and reproducibility-oriented benchmarking framework for vibration-based FD. The framework maps heterogeneous datasets into a unified recording abstraction and standardizes windowing, feature extraction, experiment configuration, and reporting. The benchmark supports grouped split strategies such as Leave-One-Run-Out or Leave-One-Recording-Out, so that all segments derived from a single acquisition remain within one partition. This design aims to reduce hidden correlations between training and test data and to provide a more defensible basis for model comparison. The implementation is available online (Knap & Jachymczyk, 2026).

The current benchmark definition includes:

- six cross-domain scenarios, covering shifts in load, acquisition frequency, fault family and severity, operating condition, damage provenance, and bearing identity;
- two datasets, Case Western Reserve University (CWRU) and Paderborn University (PU);
- comparison across ML models trained on condition indicators (CIs);
- comparison across repeated-seed DL models trained on raw vibrations;
- a design that exposes the trade-off between runtime and result quality.

The study was conducted to answer the following research questions:

- **RQ1:** What benchmark design is needed in vibration-based FD to prevent data leakage and enable meaningful comparison across studies?
- **RQ2:** How well do representative ML and DL baselines generalize under realistic cross-domain shifts in vibration data?

2. METHODOLOGY

This study evaluates cross-domain bearing fault classification under a fixed, leakage-safe benchmark protocol. Each benchmark scenario defines a source-domain training set and a target-domain test set at the recording level, and the same evaluation procedure is applied to both ML and DL baselines.

2.1. Scenario Construction and Benchmark Objective

Each scenario s is defined by a pair of metadata queries: one specifying the source-domain training split and one specifying the target-domain test split. These queries are applied to complete recordings rather than to pre-extracted windows, yielding two disjoint recording sets, $\mathcal{R}^{(s)}_{\text{train}}$ and $\mathcal{R}^{(s)}_{\text{test}}$. Fixed-size overlapping windows are extracted only after this split has been established.

For scenario s, a model f_θ is trained on source windows derived from $\mathcal{R}^{(s)}_{\text{train}}$ and evaluated on target windows derived from $\mathcal{R}^{(s)}_{\text{test}}$:

$$\hat{\theta}_s = \arg\min_{\theta} \mathcal{L}_{\text{train}}\left(\mathcal{D}^{(s)}_{\text{train}}; \theta\right), \tag{1}$$

$$\text{MacroF1}_s = \text{MacroF1}\left(f_{\hat{\theta}_s}\left(X^{(s)}_{\text{test}}\right), y^{(s)}_{\text{test}}\right). \tag{2}$$

This formulation makes the intended transfer setting explicit and reusable across datasets. In the benchmark considered here, the shifted factor depends on the scenario and includes changes in load, measurement setup, fault instance, operating condition, damage provenance, or bearing identity. The objective is therefore not only to measure classification accuracy, but also to assess how well a method trained under one set of conditions generalizes to another.

2.2. Leakage-Safe Evaluation Protocol

Preventing train–test leakage is a primary design requirement of the benchmark. Each signal acquisition is treated as an indivisible recording, so all windows derived from a given recording are assigned entirely to either the training split or the test split. This is especially important in vibration diagnostics, where overlapping windows from the same acquisition are strongly correlated and can otherwise lead to overly optimistic performance estimates.

The split is therefore fixed before any window extraction, feature computation, or neural input construction takes place. Any optional augmentation is applied only to the training split. In the ML pipeline, preprocessing is fitted on source training data and then applied to held-out target windows through a standard pipeline. Hyperparameter optimization is also restricted to the source training split. Model selection is based on macro-F1 estimated from source-side validation only, while the target/test split is reserved exclusively for final evaluation.

2.3. Benchmark Pipelines

The framework represents each loaded signal acquisition as a `Recording` object containing the signal array and associated metadata, such as class label, sampling frequency, channel information, and operating-condition descriptors when available. Collections of recordings are organized into a `Dataset` object, which also exposes a tabular metadata view used to define scenario-specific train/test splits.

Window extraction is implemented lazily through dataset views, so overlapping windows can be exposed without materializing redundant copies of the underlying signal in memory. The same principle is used for transformed representations, which keeps the framework practical for larger datasets and allows the same benchmark logic to be reused across ML and DL pipelines.

The ML pipeline extracts engineered time-domain and frequency-domain indicators from fixed windows and trains classical classifiers using Bayesian hyperparameter optimization. The DL pipeline trains neural models directly on windowed vibration signals, either in raw form or with spectral representations constructed within the model. Although the representations differ, both pipelines follow the same overall procedure:

1. resolve the experiment configuration,
2. load the dataset and construct the scenario-specific source and target splits,
3. extract overlapping windows from the selected recordings,
4. build the required feature or tensor representation,
5. train and tune candidate models on the source split,
6. evaluate the selected model on the held-out target split, and
7. serialize configuration, predictions, and summary metrics.

This shared execution flow ensures that differences between ML and DL results are attributable to the modeling approach rather than to changes in the underlying benchmark protocol.

2.4. Evaluation Metrics

The primary evaluation metric is macro-F1. For a problem with C classes, it is defined as

$$\text{MacroF1} = \frac{1}{C}\sum_{c=1}^{C} \text{F1}_c, \tag{3}$$

with

$$\text{F1}_c = \frac{2\,\text{Precision}_c\,\text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} = \frac{2TP_c}{2TP_c + FP_c + FN_c}. \tag{4}$$

where $c \in \{1, \ldots, C\}$ denotes one of the considered classes. Precision_c and Recall_c denote the precision and recall for class c, respectively. The terms TP_c, FP_c, and FN_c denote the numbers of true positives, false positives, and false negatives for class c.

Macro-F1 is preferred because class distributions in bearing diagnosis are often imbalanced, and accuracy alone may there-

fore overstate practical performance. By assigning equal weight to each class, macro-F1 gives a more informative summary of performance across both frequent and infrequent fault categories. Balanced accuracy and standard accuracy are reported as supporting metrics.

2.5. Final Reporting Protocol

The benchmark includes three computational settings: `quick`, `normal`, and `long`. These settings share the same scenario definitions but differ in model breadth, overlap settings, and optimization or training budget. In this study, their role is mainly practical: they were used during benchmark development for implementation checks, initial tuning, and candidate shortlisting.

Final reporting is intentionally conservative and differs slightly between the two pipeline families. For each scenario, both ML and DL models were tested under three computational settings. For final evaluation were selected models trained using settings that achieved best classification results. For ML, the `long` setting serves as the main reporting configuration because it provides the most complete search space and the most stable overall comparison across scenarios. For DL, single-run results are used only to identify promising configurations. The final DL conclusions are then based on repeated-seed follow-up of the shortlisted settings. In the experiments reported here, this follow-up uses three seeds, $\{41, 42, 43\}$, and its results take precedence over single-run DL outcomes whenever they lead to different conclusions.

2.6. Configuration and Traceability

The benchmark uses Hydra for configuration composition. Root configurations define dataset, model, runtime, and training defaults, while scenario-specific task configurations define the source and target domains together with task-level signal-processing parameters. This makes the benchmark protocol explicit and allows the same runner to be reused across datasets and transfer scenarios.

For each run, the resolved configuration is serialized together with run metadata, training logs, and evaluation artifacts such as summary metrics and predictions. As a result, every reported number can be traced back to a specific scenario definition and configuration state.

3. EXPERIMENTAL SETUP

3.1. Datasets and Scenarios

The benchmark uses two public bearing datasets: the Case Western Reserve University dataset (CWRU) (Smith & Randall, 2015b) and the Paderborn University dataset (PU) (Lessmeier, Kimotho, Zimmer, & Sextro, 2016). For CWRU, only the drive-end vibration channel is used. For PU, the benchmark uses the `HostService` raster, corresponding to the high-frequency vibration signal. All scenarios are defined in advance and kept fixed throughout the study.

Table 1 summarizes the six cross-domain scenarios used in the paper. For CWRU, the benchmark covers transfer across load, measurement setup, and fault instance. For PU, it covers transfer across operating condition, damage provenance, and bearing identity. Depending on the scenario, the label space contains either four classes or a reduced three-class subset.

3.2. Signal Processing and Windowing

The benchmark operates directly on the original recordings loaded from the dataset files, without resampling or denoising. Window extraction is performed only after the train/test split has been fixed, so windows from the same recording cannot appear in both partitions.

Window size is dataset-specific and fixed across the reported experiments. For CWRU, windows contain 2048 samples. For PU, windows contain 8192 samples. Test overlap is fixed at 50% throughout. Training overlap depends on the computational setting: it is 75% for the `long` setting, 50% for the `normal` setting, and 10% for `quick`. Since the final reporting is based on the best-performing setting for each case, all ML and most DL models use 75% training overlap in the final reporting setup. In the ML pipeline, features are standardized before classification. In the DL pipeline, windows are used as raw inputs unless a spectral representation is constructed inside the model.

3.3. ML Baselines and Feature Representation

The final ML benchmark considers four baseline classifiers: Logistic Regression, Random Forest, XGBoost, and K-Nearest Neighbors. All ML runs use a joint time-domain and frequency-domain representation computed at the window level.

The feature set combines standard time-domain indicators describing signal amplitude, dispersion, and impulsiveness with frequency-domain indicators describing the spectral center and spread. Frequency features are computed from the real Fourier spectrum view of each window. The purpose of this representation is not to exhaustively optimize handcrafted features, but to provide a strong and reproducible classical baseline for comparison with the DL pipeline. The complete feature specification is available in the accompanying implementation.

3.4. DL Baselines and Training Setup

The DL benchmark considers four baseline model families: a multilayer perceptron (MLP), a raw one-dimensional convolutional neural network (1D CNN), a fast Fourier transform (FFT)-based 1D CNN, and a short-time Fourier transform (STFT)-based two-dimensional CNN. The MLP and raw 1D CNN operate directly on windowed vibration signals, with

Table 1. Cross-domain scenarios used in the benchmark.

Dataset	Scenario	Shift factor	Source domain	Target domain	Classes
CWRU	Cross-Load	Load / speed	`12DriveEndFault` and `NormalBaseline`, 12 kHz, train rpm in {1797, 1772, 1750}	same acquisition setup, test rpm = 1730	4
CWRU	Cross-Sampling Frequency (Cross-FS)	Measurement setup	`12DriveEndFault`, 12 kHz, seeded fault families for training	`48DriveEndFault`, 48 kHz	3
CWRU	Cross-Fault Instance	Fault instance	`12DriveEndFault`, 12 kHz, fault families `0.007-`⋆ and `0.014-`⋆	same setup, held-out `0.021-`⋆ fault families	3
PU	Cross-Operating Condition	Operating condition	`HostService`, train operating conditions `N15_M01_F10`, `N15_M07_F04`, `N15_M07_F10`	`HostService`, test operating condition `N09_M07_F10`	4
PU	Cross-Damage Provenance	Damage provenance	`HostService`, artificial faults + healthy reference bearings `K002--K006`	`HostService`, lifetime faults + healthy reference bearing `K001`	3
PU	Cross-Bearing Instance	Bearing identity	`HostService`, all bearings except held-out identities	`HostService`, held-out bearings `K001`, `KA30`, `KI21`, `KB23`	4

the MLP using flattened inputs. The FFT and STFT models construct spectral representations within the model pipeline.

All DL models are trained with cross-entropy loss with label smoothing and AdamW optimization. The main training budget is fixed across scenarios, with 50 epochs used in the final comparisons. Learning-rate scheduling is applied during training, and standard stabilization measures such as gradient clipping are used throughout. The final reported DL configurations are selected at the scenario level and validated through repeated-seed follow-up, as described below. Table 2 summarizes the final scenario-level reporting configurations used in the paper.

Table 2. Final reporting configurations used in the paper. ML results are reported from the final long-setting benchmark. DL results are reported from the shortlisted scenario-specific setting validated by repeated-seed follow-up.

Dataset	Scenario	Final ML	Final DL
CWRU	Cross-Load	Long / XGBoost	Long / 1D CNN
CWRU	Cross-FS	Long / XGBoost	Normal / 2D STFT CNN
CWRU	Cross-Fault Instance	Long / Logistic Regression	Long / MLP
PU	Cross-Operating Condition	Long / XGBoost	Normal / 1D CNN
PU	Cross-Damage Provenance	Long / Logistic Regression	Long / 2D STFT CNN
PU	Cross-Bearing Instance	Long / Logistic Regression	Long / 1D CNN

3.5. Model Selection and Final Reporting

Macro-F1 is used as the primary model-selection and reporting metric, with balanced accuracy and standard accuracy reported as supporting measures.

For ML, hyperparameter tuning is performed on source training data only, using Bayesian optimization with five-fold cross-validation and macro-F1 as the selection criterion. The final ML results are reported from the most comprehensive benchmark setting, which provides the broadest model search and the most stable overall comparison across scenarios.

For DL, candidate model and configuration combinations are first compared under the common six-scenario benchmark. Because several single-run DL outcomes proved sensitive to initialization, final DL reporting is based on repeated-seed follow-up of the shortlisted configurations. Each shortlisted DL setting is rerun with three random seeds, $\{41, 42, 43\}$, and these repeated-seed results are used for the final DL conclusions whenever they differ from the corresponding single-run outcome.

3.6. Implementation Details

The benchmark is implemented in Python, with scikit-learn used for ML baselines and PyTorch used for DL baselines. Experiments were run in a fixed archived software environment on a workstation equipped with an Intel Core i7-11700K CPU, 64 GiB RAM, and an NVIDIA GeForce RTX 3080 GPU with 10 GiB memory. The released code and configuration archive provide the exact implementation details needed to reproduce the reported runs.

4. RESULTS

4.1. Model-Wise Comparison Across Scenarios

Figure 1 compares the performance of the baseline model families across the six benchmark scenarios. The figure highlights two points. First, scenario difficulty differs substantially: CWRU Cross-Load is comparatively easy for both ML and DL, whereas CWRU Cross-FS and PU Cross-Bearing Instance remain much more challenging. Second, no single model family dominates uniformly across all scenarios, which supports reporting final conclusions at the scenario level rather than relying on a single aggregate ranking.

4.2. Final ML Results Across Scenarios

Table 3 reports the final ML results under the long benchmark setting. The ML picture is comparatively stable: the final ranking can be read directly from the long-setting benchmark without additional follow-up. XGBoost is the strongest model for CWRU Cross-Load, CWRU Cross-FS, and PU Cross-Operating Condition, whereas Logistic Regression performs best for CWRU Cross-Fault Instance, PU Cross-Damage Provenance, and PU Cross-Bearing Instance.

The scenarios differ substantially in difficulty. CWRU Cross-Load is nearly saturated for ML, with XGBoost reaching 0.994 macro-F1. By contrast, CWRU Cross-FS remains the hard-

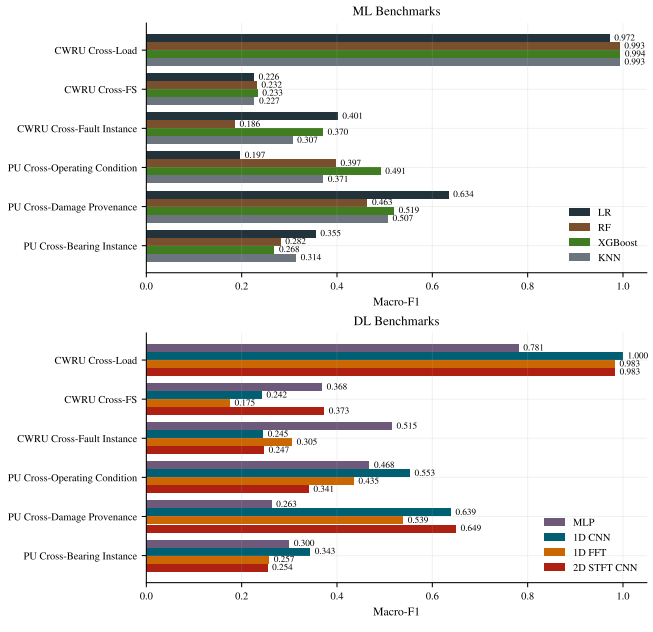

Figure 1. Macro-F1 of the baseline model families across the six benchmark scenarios. Top: ML baselines. Bottom: DL baselines.

est ML scenario, with the best result reaching only 0.233 macro-F1, indicating that transfer across acquisition regime is considerably more difficult than transfer across load alone. Among the PU scenarios, PU Cross-Damage Provenance is the strongest ML case, while PU Cross-Bearing Instance remains clearly more challenging. For ML, Table 3 reports the training time of the final long-setting run.

Table 3. Final ML results by scenario. Results are reported from the long benchmark setting.

Scenario	Best model	Macro-F1	Train time [s]
CWRU Cross-Load	XGBoost	0.994	29.1
CWRU Cross-FS	XGBoost	0.233	19.0
CWRU Cross-Fault Instance	Logistic Regression	0.401	65.1
PU Cross-Operating Condition	XGBoost	0.491	593.0
PU Cross-Damage Provenance	Logistic Regression	0.634	1565.1
PU Cross-Bearing Instance	Logistic Regression	0.355	2200.6

4.3. Final DL Results with Repeated-Seed Validation

Table 4 reports the final DL results after repeated-seed validation. Unlike ML, the final DL reporting setup is scenario-dependent rather than tied to a single global setting. The long setting is retained for CWRU Cross-Load, CWRU Cross-Fault Instance, PU Cross-Damage Provenance, and PU Cross-Bearing Instance, whereas the normal setting is retained for CWRU Cross-FS and PU Cross-Operating Condition, because it provides a more efficient setup for these scenarios.

Repeated-seed follow-up is important because several single-run DL conclusions were optimistic. The most important correction concerns PU Cross-Damage Provenance: the single-run comparison suggested that the quick STFT configuration was strongest, but repeated-seed evaluation does not support that conclusion. Instead, the long STFT configuration yields the more reliable final result. More generally, the repeated-seed means are consistently lower than or equal to the corresponding single-run reference values, showing that single-run DL snapshots can overstate final performance. For DL, Table 4 reports the mean training time across the repeated-seed follow-up runs used for final validation. It also reports the mean Macro-F1 score across the three considered seeds, together with the standard deviation.

Table 4. Final DL results by scenario after repeated-seed validation. Reported values are mean $\pm$ standard deviation over three seeds. Training time is the mean wall-clock training time across the same runs.

Scenario	Final DL model	Macro-F1	Mean train time [s]
CWRU Cross-Load	1D CNN	1.000 ± 0.000	97.5
CWRU Cross-FS	STFT CNN	0.363 ± 0.013	34.4
CWRU Cross-Fault Instance	MLP	0.512 ± 0.008	13.1
PU Cross-Operating Condition	1D CNN	0.537 ± 0.022	3041.4
PU Cross-Damage Provenance	STFT CNN	0.637 ± 0.050	3001.6
PU Cross-Bearing Instance	1D CNN	0.326 ± 0.008	7084.1

The final DL results show both strong performance and meaningful sensitivity. CWRU Cross-Load is effectively saturated, with negligible variation across seeds. CWRU Cross-Fault Instance is also relatively stable after shortlisting the long MLP configuration. In contrast, PU Cross-Damage Provenance and PU Cross-Operating Condition remain more variable, changing only the random seed can lead to differences of up to 5%, as observed for the STFT model. This indicates that good DL performance under domain shift does not necessarily imply robust reproducibility across seeds.

4.4. Cross-Scenario Analysis

The benchmark reveals a clear hierarchy of transfer difficulty. CWRU Cross-Load is the easiest scenario for both pipelines, suggesting that transfer across load and speed is relatively mild in comparison with the other shifts considered here. At the opposite end, CWRU Cross-FS and PU Cross-Bearing Instance remain the most challenging settings, showing that transfer across acquisition regime and physical bearing identity is substantially harder than transfer across load.

The ML and DL pipelines also behave differently across scenarios. ML is easier to summarize under a single final reporting setting and is comparatively stable at the protocol level, but its absolute performance remains limited in the harder cross-domain cases. DL achieves higher macro-F1 in five of the six scenarios, but this stronger performance comes with greater sensitivity to configuration choice and seed variation. PU Cross-Bearing Instance is particularly informative in this respect: it remains difficult for both pipelines and is the only scenario in which the final ML result exceeds the final DL result.

An important outcome of the benchmark is that several scenario-level results remain low in absolute terms, indicating that these transfer settings are genuinely difficult rather than already solved. For ML, the best final macro-F1 is only 0.233 on CWRU Cross-FS, 0.355 on PU Cross-Bearing Instance, and 0.401 on CWRU Cross-Fault Instance. For DL, repeated-seed validation still yields only 0.363 ± 0.013 on CWRU Cross-FS and 0.326 ± 0.008 on PU Cross-Bearing Instance. Even the stronger non-saturated cases, such as PU Cross-Operating Condition (0.537 ± 0.022) and PU Cross-Damage Provenance (0.637 ± 0.050), should therefore be interpreted as partial transfer rather than near-ceiling performance. This observation does not weaken the benchmark; on the contrary, it highlights its value, since only genuinely challenging scenarios can reveal where current methods still fail under a leakage-safe protocol.

Finally, the benchmark highlights that different types of domain shift should not be conflated. Load transfer, acquisition-frequency transfer, operating-condition transfer, provenance transfer, and bearing-identity transfer do not induce the same level of difficulty. A model that appears strong on one type of shift does not necessarily generalize equally well to another. This is precisely the role of the benchmark: to separate these transfer regimes under a common leakage-safe protocol and to make scenario-level conclusions explicit.

5. CONCLUSION

This work introduced a leakage-safe and reproducible benchmark for vibration-based bearing fault diagnosis under cross-domain shift. The benchmark evaluates both ML and DL baselines across six fixed source–target scenarios on the CWRU and PU datasets, while enforcing recording-level train/test separation to prevent leakage between correlated windows.

The reported results show that scenario difficulty is highly heterogeneous: some shifts, such as CWRU Cross-Load, are nearly saturated, whereas acquisition-regime, provenance, and bearing-identity transfer remain substantially more challenging. Under the final reporting protocol, ML is most consistently summarized by the long benchmark setting, while DL retained setting depends on the scenario and may be either long or normal. Additionally, DL models required repeated-seed follow-up to support reliable final conclusions by showing models consistence across different random initializations. The benchmark provides a reproducible basis for comparing cross-domain bearing diagnosis methods and highlights the importance of scenario-level evaluation rather than relying on aggregate performance alone.

Overall, the proposed benchmark establishes a practical framework for assessing cross-domain bearing diagnosis under realistic transfer conditions. At the same time, the results highlight that stronger benchmark settings do not automatically translate into proportionally better or more reliable conclusions, especially for DL. Future benchmark use should therefore balance performance gains against computational cost and should treat repeated-seed follow-up as an important part of reliability-oriented DL evaluation.

Beyond the results reported here, the benchmark demonstrates value as a reusable experimental framework. It allows multiple comparisons, analyses, and validation procedures to be conducted in a unified and reproducible manner, without repeated manual implementation for each new scenario. This greatly reduces the effort required to design and execute cross-domain studies, while at the same time improving consistency, transparency, and extensibility. Consequently, the benchmark should be viewed not only as a means of obtaining the present results, but as an enabling infrastructure for more systematic future research in vibration-based fault diagnosis and cross-domain bearing diagnosis.

5.1. Limitations and Future Work

The present study has several limitations. First, the benchmark is yet limited to fault diagnosis using two public bearing datasets and six fixed cross-domain scenarios, so the conclusions should be interpreted as evidence for these specific transfer settings rather than as a universal characterization of vibration-based PdM. Second, the final ML results are based on single benchmark runs, whereas repeated-seed follow-up was applied only to shortlisted DL configurations. Third, although the benchmark already covers several cross-domain scenarios under a recording-level leakage-safe protocol, its scope can be further broadened to include additional deployment-relevant conditions such as sensor aging, background noise variation, missing channels, and long-term temporal drift.

Future work should extend the benchmark both empirically and methodologically. On the empirical side, the scenario inventory can be expanded to additional datasets, sensor positions, and more realistic industrial transfer regimes. Methodologically, the benchmark can be extended with stronger evaluation procedures, including repeated-seed analysis, statistical significance testing, and additional robustness criteria. Such an extended protocol could then be used to evaluate stronger transfer-learning and domain-adaptation methods against the current ML and DL baselines.

Acknowledgment. This work was supported by the National Science Centre, Poland (NCN), under the PRELUDIUM 24 grant 2025/57/N/ST8/03313.

REFERENCES

Abburi, H., Chaudhary, T., Ilyas, S. H. W., Manne, L., Mittal, D., Williams, D., ... Veeramani, B. (2023). A closer look at bearing fault classification approaches. In *Annual conference of the phm society* (Vol. 15). doi: 10.36001/phmconf.2023.v15i1.3473

Apicella, A., Isgrò, F., & Prevete, R. (2025, aug). Don't push the button! exploring data leakage risks in machine learning and transfer learning. *Artificial Intelligence Review*, *58*(11). doi: 10.1007/s10462-025-11326-3

Chen, X., Yang, R., Xue, Y., Huang, M., Ferrero, R., & Wang, Z. (2023). Deep transfer learning for bearing fault diagnosis: A systematic review since 2016. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–21. doi: 10.1109/TIM.2023.3244237

Hendriks, J., Dumond, P., & Knox, D. (2022). Towards better benchmarking using the cwru bearing fault dataset. *Mechanical Systems and Signal Processing*, *169*, 108732. doi: https://doi.org/10.1016/j.ymssp.2021.108732

Knap, P., & Jachymczyk, U. (2026). *pdm-bench: Benchmark for predictive maintenance.* `https://github.com/1Sensor/pdm-bench`.

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, *138*, 106587. doi: https://doi.org/10.1016/j.ymssp.2019.106587

Lessmeier, C., Kimotho, J. K., Zimmer, D., & Sextro, W. (2016). 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 *Phm society european conference* (Vol. 3). doi: 10.36001/phme.2016.v3i1.1577

Matania, O., Cohen, R., Bechhoefer, E., & Bortman, J. (2024). Test-training leakage in evaluation of machine learning algorithms for condition-based maintenance. In *Proceedings of the phm society european conference (phme 2024)* (p. 13). Prague, Czech Republic.

Neupane, D., Bouadjenek, M. R., Dazeley, R., & Aryal, S. (2025). Data-driven machinery fault diagnosis: A comprehensive review. *Neurocomputing*, *627*, 129588.

Rosa, R. K., Braga, D., & Silva, D. (2024). *Benchmarking deep learning models for bearing fault diagnosis using the cwru dataset: A multi-label approach.* Retrieved from `https://arxiv.org/abs/2407.14625`

Smith, W. A., & Randall, R. B. (2015a). Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. *Mechanical Systems and Signal Processing*, *64-65*, 100-131. doi: https://doi.org/10.1016/j.ymssp.2015.04.021

Smith, W. A., & Randall, R. B. (2015b). Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. *Mechanical systems and signal processing*, *64*, 100–131.

Vieira, J. P., Bauler, V. A., Rosa, R. K., & Silva, D. (2026). *Towards a more realistic evaluation of machine learning models for bearing fault diagnosis.* Retrieved from `https://arxiv.org/abs/2509.22267`

Zhao, Z., Li, T., Wu, J., Sun, C., Wang, S., Yan, R., & Chen, X. (2020). Deep learning algorithms for rotating machinery intelligent diagnosis: An open source benchmark study. *ISA Transactions*, *107*, 224-255.