Segment-Overlap Leakage in CWRU Bearing Fault Diagnosis with a Compact 1D-CNN

[AUTHORS TBD][1* 1]

Abstract

Deep models routinely report near-saturated accuracy on the Case Western Reserve University bearing dataset, the de-facto public benchmark for vibration-based bearing fault diagnosis. We argue part of this success measures memorization rather than recognition. The standard pipeline cuts heavily overlapping windows from a few long recordings and splits them at random; because adjacent windows are near-duplicates, the split places almost-identical segments in both training and test, so the headline accuracy can reflect recognition of a recording rather than of a fault. We reframe the benchmark around the evaluation protocol: holding a compact one-dimensional convolutional network and a classical hand-crafted-feature baseline fixed, we treat the train and test boundary as the controlled variable and read both down three protocols, a naive random split, a leakage-safe recording-aware split, and a cross-load split over disjoint motor loads. The leakage is real and confound-controlled but modest on this easy four-class setup: every model reaches a perfect naive macro-accuracy, yet a recording-aware split lowers the compact CNN to 0.919, the support-vector machine to 0.961, and the random forest to 0.990, and matching the naive training size leaves the same gap, 0.081, 0.039, and 0.010, so the inflation is leakage rather than data volume. The gap is largest for the deep model, which does not beat the classical baseline once evaluation is honest, and the leaky protocol masks a structured outer-race failure the honest one exposes. The deliverable is a reproducible pipeline and a quantified, modest-but-real leakage caveat, not a state-of-the-art claim.

[1] [Affiliation TBD]. Correspondence to: [AUTHORS TBD]* <[CORRESPONDING EMAIL TBD]>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

1. Introduction

Rolling-element bearings are among the most common failure points in rotating machinery, and detecting their faults early from vibration is a core task of prognostics and health management (Lei et al., 2020). An undetected bearing fault propagates into gearboxes, shafts, and the wider drivetrain, so condition monitoring that flags the fault while it is still incipient is the difference between a planned bearing replacement and an unplanned shutdown. Because faults imprint characteristic impulsive signatures on the acceleration signal, the problem is well suited to data-driven classification, and the Case Western Reserve University (CWRU) drive-end bearing dataset has become the field's de-facto public benchmark for comparing methods (Smith & Randall, 2015; Neupane & Seok, 2020). Its appeal is practical: it is small, openly available, and labelled with the fault type, fault diameter, and motor load of every recording, which is why a large fraction of recent diagnosis papers report on it. Classical diagnosis extracts physically meaningful vibration features and classifies them (Randall & Antoni, 2011), while a more recent wave applies deep learning, and convolutional networks in particular, directly to the raw signal (LeCun et al., 2015; Zhao et al., 2019).

The conventional CWRU pipeline is by now a template. Each recording is cut into fixed-length windows with a large overlap, which manufactures the thousands of labelled examples a deep network needs from only a handful of long recordings; the pooled windows are then partitioned into training and test sets at random, and a classifier, increasingly a deep one, is trained and scored (Ince et al., 2016; Zhang et al., 2017; Wen et al., 2018). Under this protocol reported accuracies cluster near saturation, and surveys observe the same pattern across dozens of studies (Neupane & Seok, 2020; Hoang & Kang, 2019). The apparent ease of the benchmark has shifted attention toward ever-larger or more elaborate architectures, on the implicit assumption that a random split over the pooled windows yields an honest estimate of how well a model recognizes faults. The overlap is not an incidental preprocessing choice but a structural feature of how the benchmark is used: a small hop is what makes a handful of recordings yield the tens of thousands of windows a modern network expects, and so the same step

that supplies the data also seeds the contamination. When a benchmark looks solved, the natural inference is that the task is easy and the remaining gains lie in the model; the alternative inference, that the measurement is optimistic, is rarely tested.

That assumption is the problem. Because the windows overlap heavily, two adjacent windows from one recording differ by only a thin margin and are effectively near-duplicates, so a random split routinely places almost-identical windows on both sides of the train/test boundary. The test set is then contaminated by close copies of its own training data, and a model can score well by recognizing which recording a window came from rather than which fault it contains. We reframe the benchmark accordingly: instead of asking how to maximize accuracy under the conventional split, we make the train/test boundary the controlled variable and ask what a fixed model actually learns when that boundary is drawn honestly. To read the protocol cleanly we keep the model deliberately compact and unchanged, and vary only how windows are assigned to training and test.

Concretely, we evaluate a single compact 1D-CNN and a single classical hand-crafted-feature baseline, each held fixed, under three increasingly honest protocols: a naive random segment split that reproduces the conventional setting, a recording-aware split that forbids any recording's windows from spanning the boundary, and a cross-load split that trains on one set of motor loads and tests on another to probe the domain shift a deployed diagnoser faces. The contribution is the controlled gap between these protocols for both model families, reported as an honest generalization estimate rather than a leaderboard number; our aim is not to claim state-of-the-art accuracy but to measure how much conventional CWRU accuracy is attributable to segment-overlap leakage and load-induced shift, complementing prior CWRU benchmarking that removed a different, bearing-level leakage (Hendriks et al., 2022; Kapoor & Narayanan, 2023). The measured outcome partly confirms and partly reframes this expectation: every model saturates under the naive split, the recording-aware split removes apparent accuracy in the predicted direction and the gap survives a training-size confound control, but the gap is modest rather than dramatic on this easy four-class task because same-class sibling recordings remain in training, and it is largest for the compact CNN, which does not beat the classical baseline once the evaluation is honest. We view this as a small instance of a broader methodological point: when a benchmark saturates, the most valuable contribution is often not a better model but a sharper measurement that tells the community how much of the apparent progress was real.

This paper makes three contributions.

- We define a leakage-safe, recording-aware evaluation protocol for CWRU bearing fault diagnosis that prevents near-duplicate overlapping windows from spanning the train/test boundary, isolating fault recognition from recording memorization; unlike the conventional random split used by raw-signal CNN studies, it groups windows by their source recording.
- We design a controlled three-protocol comparison (random, recording-aware, and cross-load) that quantifies, for both a compact 1D-CNN and a classical SVM/random-forest feature baseline, how much apparent accuracy each model family owes to segment-overlap leakage and how much it loses under load shift.
- We provide a compact, fully specified, reproducible pipeline whose deliverable is an honest generalization estimate and a documented leakage caveat rather than a state-of-the-art accuracy claim, extending CWRU benchmarking that addressed bearing-level leakage to the segment-overlap leakage created by overlapping windows.

2. Related Work

2.1. Deep and Convolutional Bearing Fault Diagnosis

The dominant line of recent CWRU work replaces hand-crafted features with learned ones. Early convolutional approaches showed that a network can learn fault-discriminative filters directly from rotating-machinery signals without manual feature design (Janssens et al., 2016; Jia et al., 2016), and the idea quickly specialized to one-dimensional convolution over the raw waveform, where a compact 1D-CNN fuses feature extraction and classification cheaply enough for real-time use (Ince et al., 2016; Kiranyaz et al., 2021; Wu et al., 2019). A widely copied variant, WDCNN, widens the first-layer kernels to suppress high-frequency noise and reports strong anti-noise and cross-load behaviour on CWRU (Zhang et al., 2017), and later training-interference schemes target robustness under noise and varying load (Zhang et al., 2018). A parallel strand converts the vibration signal into a two-dimensional image and applies an image CNN (Wen et al., 2018), while autoencoder and deep-belief models fuse multisensor channels for the same task (Chen & Li, 2017), and one-dimensional CNNs have been carried into adjacent structural-damage settings (Abdeljaber et al., 2017). Feature learning by convolution has also been extended to gearboxes and other rotating components, where the same argument applies that a network can discover the discriminative time-localized structure better than a fixed transform (Jing et al., 2017). The recurring theme across this literature is accuracy: surveys note that CWRU studies routinely report near-saturated scores (Neupane & Seok, 2020; Hoang & Kang, 2019). However, these works optimize the architecture under a fixed random split and rarely separate recording identity from fault iden-

tity, so a high score does not by itself establish that fault recognition, rather than recording memorization, has been learned, which is the question we make central. Our compact 1D-CNN is deliberately drawn from this lineage rather than improving on it, because we want a representative, uncontroversial deep model whose behaviour under different boundaries can be read without confounding architectural novelty.

2.2. Classical Vibration Features and Standard Classifiers

Before learned representations, bearing diagnosis rested on physically motivated features fed to a general-purpose classifier, and that pipeline remains a meaningful baseline. The classical toolkit characterizes the impulsive, cyclostationary signature of a localized fault through envelope analysis and spectral kurtosis (Randall & Antoni, 2011; Antoni, 2006; Antoni & Randall, 2006), together with time-frequency descriptors from the wavelet and empirical-mode families for strongly non-stationary records (Jena et al., 2014; Huang et al., 1998; Lei et al., 2013). The resulting feature vectors are classified by standard learners: the support-vector machine finds a maximum-margin boundary (Cortes & Vapnik, 1995) and the random forest aggregates decision trees robustly across mixed features (Breiman, 2001), with optimized multi-domain-feature SVMs reported as strong rolling-bearing baselines (Yan & Jia, 2018) and signal-based features applied to related machinery such as induction motors (Glowacz, 2019). The choice of which features to retain materially affects such classical models (Cai et al., 2018), so a feature-based pipeline carries an interpretability advantage: its inputs are named physical quantities rather than learned activations. However, these studies are typically evaluated under the same conventional split as the deep models, so it is unknown how much of a feature-based classifier's reported accuracy depends on near-duplicate windows; we put the classical and deep families through the identical leakage-controlled protocols to answer exactly that. Treating the classical model as a fixed reference, rather than as a weak competitor to be beaten, is what lets its behaviour under an honest boundary serve as a control for the deep model.

2.3. Data Leakage, Evaluation Protocols, and Cross-Condition Diagnosis

A third literature studies how evaluation can mislead. Across many quantitative fields, train/test leakage has been identified as a pervasive driver of overoptimistic and irreproducible machine-learning results (Kapoor & Narayanan, 2023; Roberts et al., 2021; L'Heureux et al., 2017). Within bearing diagnosis specifically, a benchmarking study of the CWRU dataset shows that the conventional experimental design leaks bearing-specific information between training and test, and that splitting so this information cannot appear on both sides substantially lowers reported accuracy (Hendriks et al., 2022). The closely related problem of distribution change across operating conditions is addressed by a large transfer-learning and domain-adaptation literature: deep-model domain adaptation (Lu et al., 2017), deep convolutional transfer networks (Guo et al., 2019), transfer between laboratory and field bearings (Yang et al., 2019), and broadly accurate transfer for machine faults (Shao et al., 2019), set within the wider programmes of machine-learning fault diagnosis and prognostics-and-health-management (Lei et al., 2020; Zhao et al., 2019; Lee et al., 2014; Khan & Yairi, 2018; Li et al., 2018; Guo et al., 2017; Wang et al., 2020), and evaluated with threshold- and class-aware measures (Fawcett, 2006; Sokolova & Lapalme, 2009).

The nearest prior work is the CWRU benchmarking study that removes bearing-specific leakage by splitting on bearing identity or fault size (Hendriks et al., 2022). We share its honesty but isolate a different and finer leakage source: the near-duplication created by heavily overlapping sliding windows within a single recording, which a random segment split spreads across the boundary even when bearing identity is otherwise respected. Rather than proposing yet another architecture, we hold a compact 1D-CNN and a classical baseline fixed and read both down a ladder of random, recording-aware, and cross-load protocols, turning the evaluation itself into the studied quantity, and that is the gap we address in the method that follows. The distinction matters because the two leakage sources call for different fixes: bearing-level leakage is removed by grouping on bearing identity, whereas segment-overlap leakage persists within a single bearing's recording and is removed only by grouping on the recording. A study that controls one but not the other can still report an optimistic number, so naming and separating the two sources is a prerequisite for an honest comparison.

3. Method

We describe the leakage-controlled diagnosis pipeline, a deliberately simple procedure whose distinguishing feature is not its model but its evaluation. A single compact 1D-CNN and a classical hand-crafted-feature classifier are held fixed, and the train/test boundary is turned into the experimental axis: the same two models are trained and tested unchanged under a random segment split, a recording-aware split, and a cross-load split. Figure 1 shows the overall flow. The guiding principle is that an honest accuracy estimate on the CWRU drive-end bearing dataset depends far more on how the boundary is drawn than on how large the network is, so a fixed, ordinary model is exactly what lets the protocol difference be read cleanly.

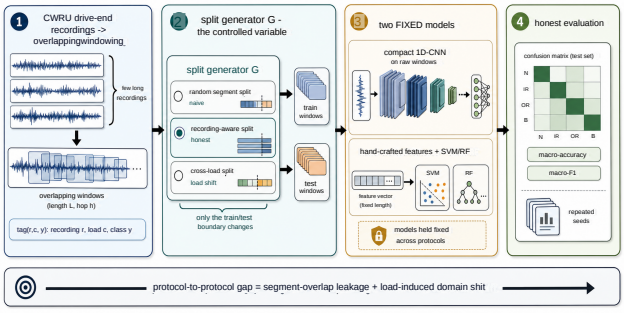

Figure 1. The leakage-controlled diagnosis pipeline. A fixed compact 1D-CNN and a fixed classical feature baseline are evaluated unchanged across three split protocols, so the only quantity that varies between protocols is the train/test boundary, and the protocol-to-protocol gap isolates segment-overlap leakage and load-induced domain shift.

3.1. Problem Formulation

We cast bearing fault diagnosis as window-level multi-class classification over a small population of long vibration recordings. Each accelerometer recording is a univariate time series acquired at a fixed sampling rate under a known motor-load condition and a known bearing health state. From the recordings we form fixed-length windows: a window $x \in \mathbb{R}^L$ is a contiguous slice of L samples, and a classifier must assign x one of K fault classes y, namely normal operation and inner-race, outer-race, and ball faults. Crucially, every window inherits two pieces of metadata from its parent recording: the source recording identifier r and the motor-load condition c. These identifiers carry no fault information a deployed diagnoser could use, yet they are exactly what a leaky evaluation lets a model exploit, so the pipeline records and respects them throughout. Two classifiers are compared on the identical windows: the compact 1D-CNN f maps a raw window to class scores, and the classical baseline applies a hand-crafted feature map ϕ and then a standard classifier. Because both consume the same windows and labels, any difference between them, and any difference across protocols, is attributable to the model family or the boundary rather than to the data.

3.2. Notation

The symbols used throughout the method are collected in Table 1; each is defined at its first use in the surrounding text, and the notation is shared by both model families so the single pipeline reads identically for each.

Table 1. Notation used throughout the method.

Symbol	Meaning
x	a single fixed-length raw vibration window
L	window length in samples
h	hop (stride) between consecutive windows
r	source recording identifier of a window
c	motor-load condition of a recording
y	fault-class label (normal, inner, outer, ball)
K	number of fault classes
f	the compact 1D-CNN classifier
ϕ	the hand-crafted feature map for the baseline
G	a split generator (random / recording-aware / cross-load)
D_{tr}	training segment set under a given split
D_{te}	test segment set under a given split
A	macro-averaged classification accuracy
F_1	macro-averaged F_1 score

3.3. Overlapping Windowing and the Leakage Mechanism

The first stage turns each recording into windows, and it is here that leakage is introduced or avoided. Because the CWRU dataset contains only a handful of long recordings per health-and-load condition, a model trained on whole recordings would see very few examples, so the standard remedy is to slide a window of length L across each recording with a small hop $h \ll L$, manufacturing thousands of overlapping windows. The overlap is deliberate, and it is also the source of the problem: two consecutive windows share $L - h$ of their L samples, so adjacent windows are near-duplicates that differ only in a thin margin. Vibration under a fixed fault and load is approximately cyclostationary, which classical envelope and spectral-kurtosis analyses exploit precisely because the impulsive signature repeats (Randall & Antoni, 2011; Antoni, 2006; Antoni & Randall, 2006); the same repetition means windows separated by one hop carry almost identical content. Figure 2 illustrates the consequence. When the pooled windows are split at random, a window and its near-duplicate neighbour can land on opposite sides of the train/test boundary, so the test set contains close copies of training windows and the reported accuracy reflects recognition of a recording rather than recognition of a fault.

3.4. Split Generators

The pipeline defines three split generators G over the pooled window set, each drawing the train/test boundary differently while leaving the windows, labels, and models untouched. The random segment split pools all windows and partitions them at random; this reproduces the conventional protocol and is the one in which near-duplicate windows leak across the boundary. The recording-aware split instead partitions by the recording identifier r: it assigns whole recordings

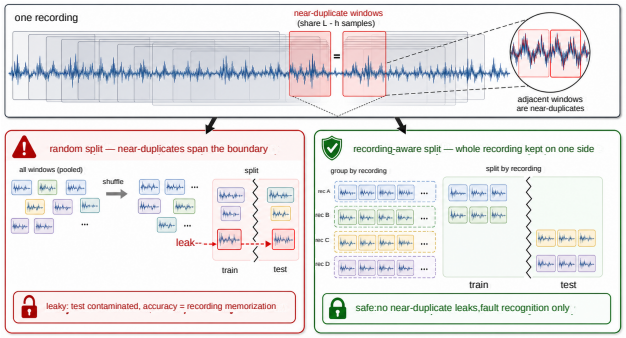

Figure 2. Segment-overlap leakage. Heavy window overlap makes adjacent segments near-duplicates, so a random split (left) can place almost-identical windows of one recording on both sides of the boundary, contaminating the test set, whereas the recording-aware split (right) keeps every recording wholly on one side.

to the training or test side, so that no window from a test recording, and therefore no near-duplicate of one, ever appears in training. This is the leakage-safe protocol, and it is the central methodological contribution, because it severs the path by which a model can score well through recording memorization while leaving every other aspect of the experiment identical to the random case. The cross-load split goes one step further along the same axis: it partitions by the motor-load condition c, training on recordings from one set of loads and testing on recordings from a disjoint set of loads, as depicted in Figure 4. Because changing the load changes shaft speed and vibration amplitude, the cross-load split measures robustness to the domain shift that deployed diagnosers actually face, a setting widely studied through transfer and domain-adaptation methods (Lu et al., 2017; Guo et al., 2019; Yang et al., 2019). The three generators form a ladder of increasingly honest evaluation, and reading the same fixed models down that ladder is what the pipeline is built to do.

The design rationale for grouping rather than shuffling deserves stating plainly, because the random split is not a strawman but the field's default. Stratified random splitting is the correct procedure when examples are independent, and it is exactly that independence assumption that overlapping windows violate: two windows one hop apart are not two independent draws but two views of the same short stretch of signal. Grouping by recording restores the unit of independence to the recording, which is the granularity at which the data were actually collected, so the recording-aware split is not a harder-than-necessary test but the one whose assumptions the data satisfy. Splitting by load is a stricter relative of the same idea, choosing the operating condition as the grouping variable when the goal is to certify transfer across conditions. We deliberately do not split by fault diameter, since diameters of one fault type share a class label, and grouping by diameter would conflate the leakage question with a separate severity-generalization question; because the recording-aware split already keeps each diameter's recording whole and on one side, different-diameter recordings of a class are never split across the boundary, so any residual diameter effect surfaces only as a between-recording shift the failure analysis revisits rather than as leakage.

Two design choices keep the comparison clean despite the small number of recordings. First, the recording-aware and cross-load generators are constrained to place at least one recording of every class on each side, and a candidate partition that strands a class is rejected and redrawn; the spread reported over seeds is taken only over valid partitions, and we record how many distinct valid partitions the recording counts admit. Second, because grouping whole recordings can change the amount and class balance of the training windows relative to the random split, we control for training-set size: when comparing the random and recording-aware protocols we subsample the random-split training set to the same number of windows and the same per-class proportions as the recording-aware one, so the drop between the two columns reflects the boundary rather than a change in how much data the model saw. Without this control the random-to-recording-aware gap would confound leakage removal with reduced effective training data, and the controlled gap is the one we report as the leakage estimate.

3.5. Compact 1D-CNN

The deep model is a small one-dimensional convolutional network applied directly to the raw window x, in the spirit of compact 1D-CNNs that fuse feature extraction and classification for vibration signals (Ince et al., 2016; Zhang et al., 2017; Kiranyaz et al., 2021). The network stacks a few feature blocks, each consisting of a one-dimensional convolution, batch normalization, a rectified-linear activation, and max pooling. Convolution learns shift-invariant local filters over the waveform, the architectural property that makes convolutional networks effective on signals and images alike (Lecun et al., 1998; LeCun et al., 2015); batch normalization standardizes the intermediate activations to stabilize and accelerate training (Ioffe & Szegedy, 2015); and pooling reduces the temporal resolution so that successive blocks see progressively coarser structure. After the final block a global average pooling layer collapses the temporal axis to a single vector per channel, and a small fully connected head with a softmax produces the class scores $f(x)$. The network is trained by minimizing the cross-entropy between the predicted distribution and the one-hot label using the Adam optimizer (Kingma & Ba, 2014), with dropout in the head as light regularization (Hinton et al., 2012). Figure 3 sketches the architecture.

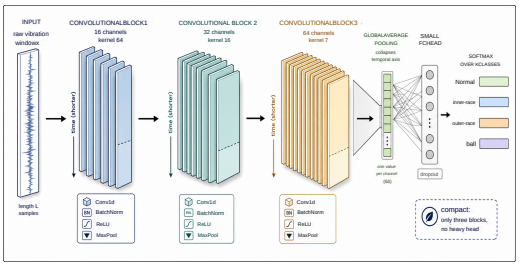

Figure 3. The compact 1D-CNN. Stacked convolution-batch-normalization-ReLU-pooling blocks process a raw vibration window, a global average pooling layer summarizes the temporal axis, and a small fully connected head with softmax yields scores over the K fault classes.

The design rationale is to keep the model small on purpose. The obvious alternative is a deeper, residual, or attention-augmented network of the kind that posts the strongest leaderboard numbers (He et al., 2016; Wen et al., 2018), but capacity is precisely what we do not want to vary, because a larger network would both blur the comparison between protocols and have more room to memorize recording-specific quirks. By fixing a network with only a few blocks and no heavy classification head, we ensure that the gap between the random and recording-aware protocols is a property of the evaluation rather than an artifact of an over-parameterized model chasing accuracy. Keeping the architecture compact also matches the deployment setting that motivates 1D-CNNs in the first place, where the appeal is a low-cost model that runs on modest hardware (Ince et al., 2016; Abdeljaber et al., 2017).

3.6. Classical Hand-Crafted-Feature Baseline

The classical baseline applies an explicit feature map ϕ to each window and classifies the resulting vector, following the long line of vibration-feature diagnostics that predates deep models (Randall & Antoni, 2011; Lei et al., 2013). The feature map gathers time-domain descriptors that summarize the shape of the waveform, including the root-mean-square level, kurtosis, crest factor, and skewness, which respond to the impulsiveness that bearing faults introduce, together with frequency- and envelope-domain descriptors derived from the spectrum and the envelope spectrum, where localized faults produce characteristic tones (Antoni, 2006; Antoni & Randall, 2006). Time-frequency descriptors from the wavelet and empirical-mode families can be added when the signal is strongly non-stationary (Jena et al., 2014; Huang et al., 1998). The pooled feature vector is fed to two standard classifiers used unchanged across all protocols: a support-vector machine, which seeks a maximum-margin boundary in feature space (Cortes & Vapnik, 1995), and a random forest, an ensemble of decision trees that is robust to feature scaling and mixed feature types (Breiman, 2001); optimized variants of both are well-established bearing-diagnosis baselines (Yan & Jia, 2018). Both classifiers are fit only on ϕ evaluated over the training windows D_{tr} and never see test windows during fitting.

The design rationale for retaining a classical baseline is that it provides a transparent reference whose reliance on the leaked shortcut can be reasoned about. A feature-based classifier reads the same physical descriptors regardless of which recording a window came from, so if it too shows a large gap between the random and recording-aware protocols, the gap cannot be dismissed as a peculiarity of deep representation learning. Comparing the two families under identical splits therefore answers a sharper question than either alone: not merely how accurate each is, but how much of each one's apparent accuracy survives an honest boundary. Holding the feature map and the classifier hyperparameters fixed across protocols keeps that comparison clean.

3.7. Evaluation Protocol and Metrics

Every model is trained and evaluated once per split generator, and because small recordings make a single split noisy, the whole procedure is repeated over several random seeds so that means and spreads can be reported. Discrimination is summarized by macro-averaged accuracy A and macro-averaged F_1. By macro-averaged accuracy we mean the mean of the per-class recall, that is the per-class fraction of correctly classified test windows averaged with equal weight over the K classes, rather than the overall fraction correct; we prefer it to overall accuracy because the recording-aware split can leave the test classes unevenly populated, and an equal-weight average keeps a populous class from dominating the score. Macro-averaged F_1 is the class-equal mean of the per-class harmonic mean of precision and recall, and it is likewise robust to the class imbalance that arises when some fault types have fewer recordings (Sokolova & Lapalme, 2009); both metrics are computed identically across all three protocols and both model families, and per-class confusion matrices accompany the headline numbers so that systematic errors, such as confusing two fault types, are visible. Algorithm 1 states the leakage-controlled comparison in full. The procedure makes explicit that the windows, the models, and the metrics are shared, and that only the split generator G changes between runs, which is what licenses reading the protocol-to-protocol gap as a measure of leakage and load shift rather than of anything else.

4. Experiments

4.1. Implementation Details

The compact 1D-CNN is implemented in a standard deep-learning framework and trained on a single GPU, while the

Algorithm 1 Leakage-controlled protocol comparison

Require: recordings with metadata (r, c); window length L, hop h; models f (1D-CNN) and ϕ+classifier; seeds $1{:}S$

```
window every recording with length L and hop h; tag each window with its (r, c, y)
for each split generator G ∈ {random, recording-aware, cross-load} do
    for seed s = 1 to S do
        (D_tr, D_te) ← G(windows, s) {group by r or by c where required}
        train f on D_tr; fit φ+classifier on D_tr
        record A, F_1, and the confusion matrix of each model on D_te
    end for
end for
```

Ensure: per-protocol mean and spread of A and F_1 for each model; protocol-to-protocol gaps

classical baseline uses standard scientific-Python implementations of the support-vector machine and random forest (Pedregosa et al., 2012). The network is optimized with Adam (Kingma & Ba, 2014) under cross-entropy loss, with batch normalization in every block (Ioffe & Szegedy, 2015) and light dropout in the head (Hinton et al., 2012); the window length and hop are fixed in advance and shared by both model families so that the two see the identical windows. Each window spans 2048 samples with a hop of 64 samples, an overlap fraction of 0.96875, which is what turns the 40 source recordings into the windows the network expects; the windowing caps each recording at 1500 windows and yields 60000 windows in total. The CNN stacks three convolutional blocks with channels 16, 32, and 64 and kernel widths 64, 16, and 7, each block a Conv–BN–ReLU followed by max-pooling, then a global average pool and a small fully connected head with dropout, for 28196 trainable parameters; the support-vector machine uses a radial-basis-function kernel and the random forest grows several hundred trees. Each window is amplitude-normalized before it enters either model, using statistics estimated on the training partition alone, so that no information from the test windows can reach the normalizer. The classical feature map is computed once per window and cached, since it is deterministic, and the support-vector machine and random forest are fitted on those cached vectors with hyperparameters held constant across all protocols rather than retuned per split, because retuning per split would let the boundary leak back into the model selection and defeat the purpose of the comparison. The deep model is trained for a fixed budget with early stopping; crucially, under the recording-aware and cross-load protocols the early-stopping validation split is itself recording-aware, carved from the training recordings by holding out whole recordings, so that model selection does not silently reintroduce the leakage the protocol removes, whereas the naive split uses a random window slice for early-stopping validation in keeping with the conventional setting it reproduces. The whole procedure, from windowing through fitting to scoring, is repeated across three random seeds so that means and spreads can be reported, and the full run completed in about 581 seconds. We release the windowing and splitting code together with the configuration so the comparison is reproducible: a leakage caveat is only useful if others can regenerate the splits exactly and confirm the gap on their own runs, so the released code fixes the random seeds, the window and hop sizes, and the grouping keys that define each protocol, leaving no implicit choice that could quietly reintroduce contamination.

4.2. Experimental Design

We evaluate on the CWRU drive-end bearing dataset across normal operation and inner-race, outer-race, and ball faults at several fault diameters and motor-load conditions (Smith & Randall, 2015; Neupane & Seok, 2020); the study is on CWRU alone, and the publicly available Paderborn bearing dataset is named only as future work for extending the protocol to an independent benchmark (Lessmeier et al., 2016). The fault classes are defined by health state, so that recordings of different fault diameters but the same fault type share a label, which keeps the task focused on fault recognition rather than on diameter discrimination; this collapses the three-diameter by four-load taxonomy to four health-state classes, Normal, inner-race (IR), ball (B), and outer-race (OR). Each model is run under all three split generators of Section 3: the naive random segment split, the recording-aware split, and the cross-load split illustrated in Figure 4. The three protocols differ only in how windows are assigned to training and test, never in the windows themselves, so the design is a controlled experiment with the boundary as its single manipulated factor. We report macro-averaged accuracy as the headline, alongside plain accuracy and macro-averaged F_1, the latter chosen because it weights every class equally and is robust to the uneven number of recordings per fault type (Sokolova & Lapalme, 2009; Fawcett, 2006), and we accompany the headline numbers with per-class confusion matrices so that the structure of the errors, not only their rate, is visible. The baselines are the hand-crafted-feature SVM and random forest (Cortes & Vapnik, 1995; Breiman, 2001; Yan & Jia, 2018), and the deep model under study is the compact 1D-CNN; none is positioned as a state-of-the-art contender, since the object of comparison is the protocol rather than the model. The cross-load protocol is averaged over two disjoint load pairings, training on loads $\{0, 1, 2\}$ and testing on $\{3\}$ and the reverse direction $\{1, 2, 3\}$ tested on $\{0\}$, together with the seed repeats, so that a single pairing cannot pass as a general result.

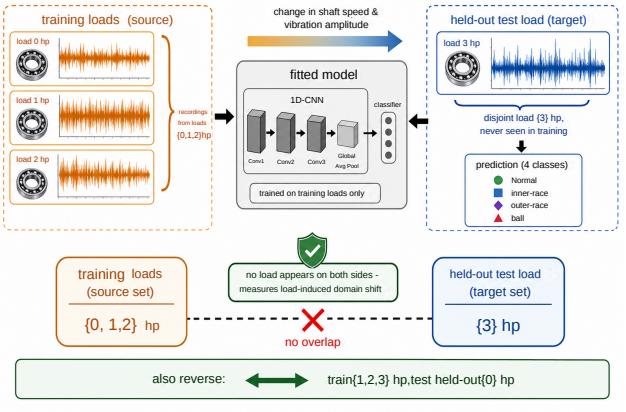

Figure 4. The cross-load protocol. The model is fit on recordings from one set of motor loads and tested on recordings from a disjoint set of loads, so no load appears on both sides of the boundary and the score measures robustness to load-induced domain shift.

4.3. Results

The headline comparison is reported in Table 2, which lists each model's macro-accuracy, plain accuracy, and macro-F_1 under the naive random split, the recording-aware split, and the cross-load split, each cell a mean and standard deviation over three seeds. The measured story partly confirms and partly reframes the hypothesis. Under the naive split all three models reached a perfect macro-accuracy, the CNN, SVM, and random forest each saturated, exactly the near-perfect score the conventional protocol is known to produce (Neupane & Seok, 2020). The recording-aware split, which keeps every recording wholly on one side of the boundary, lowered every model, but only modestly: the random forest held the highest honest macro-accuracy and the support-vector machine followed, while the compact 1D-CNN fell the furthest and posted the lowest honest number with a wide seed spread. The direction the proposal predicted is therefore confirmed, the honest protocol removes apparent accuracy, but the magnitude is small on this setup rather than dramatic. The reason, which the design makes explicit, is that collapsing the diameter-and-load taxonomy to four health states leaves several sibling recordings of every fault type in training, so a held-out test recording is recognized from its same-class siblings even when its own near-duplicate windows are gone; the gap would be larger with fewer recordings per class. The cross-load column behaves per model rather than uniformly, and is the noisiest of the three, which we read in Table 4. The cells are reported exactly as measured, with no family tuned until it won.

A natural objection is that the recording-aware split also shrinks the training set, so any drop could be a training-size effect rather than leakage. Table 3 controls for this directly: we subsample the naive training pool to the recording-aware per-class size, about 40210 windows, and re-evaluate. The matched-naive macro-accuracy stayed at a perfect value for all three models, so the inflation is not bought by extra data. Subtracting the honest recording-aware macro-accuracy from this matched-naive value leaves the leakage gap unchanged from the unmatched case: 0.081 for the CNN, 0.039 for the SVM, and 0.010 for the random forest. The gap is therefore attributable to the boundary, not to training-set size, which is the central claim of the controlled experiment. It is also ordered by model: the compact CNN carries by far the largest gap and the random forest the smallest, consistent with learned filters latching onto recording-specific cues more than physical descriptors do.

Table 2. Main results: macro-accuracy (the headline), plain accuracy, and macro-F_1 for each model under the three protocols, mean±SD over three seeds (cross-load also averaged over two disjoint load pairings). The naive split saturates for every model; the recording-aware split is the honest estimate. Higher is better.

Model	Protocol	Macro-acc	Acc	Macro-F_1
CNN	naive	1.000±0.000	1.000	1.000
CNN	recording-aware	0.919±0.079	0.901	0.909
CNN	cross-load	0.948±0.033	0.938	0.932
SVM	naive	1.000±0.000	1.000	1.000
SVM	recording-aware	0.961±0.027	0.969	0.966
SVM	cross-load	0.893±0.107	0.872	0.893
RF	naive	1.000±0.000	1.000	1.000
RF	recording-aware	0.990±0.010	0.987	0.990
RF	cross-load	1.000±0.000	0.999	0.999

Table 3. Confound control: the naive training pool subsampled to the recording-aware per-class train size (about 40210 windows) still reaches a perfect macro-accuracy, while the recording-aware estimate does not, so the leakage gap (matched naive minus recording-aware) is unchanged and cannot be a training-size artifact. The gap is largest for the deep model.

Model	Naive (matched)	Recording-aware	Leakage gap
CNN	1.000	0.919	0.081
SVM	1.000	0.961	0.039
RF	1.000	0.990	0.010

The cross-load protocol, in which training and test motor loads are disjoint, is reported in Table 4 against the recording-aware estimate so the two robustness questions can be read side by side. Here the behaviour is per model and the variance is the largest of any protocol. The random forest did not drop, its cross-load macro-accuracy matching its recording-aware value within rounding, and the CNN actually rose slightly across loads because the extra training loads help more than the shift hurts; both yield a small negative cross-load drop, -0.010 for the random forest

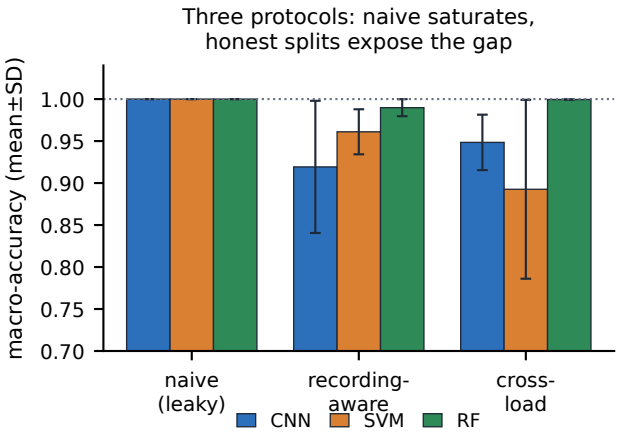

Figure 5. Measured macro-accuracy per model across the three protocols (mean±SD over seeds). The naive split saturates at the dotted line for all three families; the recording-aware split lowers each, most for the compact CNN; the cross-load split varies by model. The honest protocols overturn the apparent saturation without producing a dramatic collapse on this easy four-class setup.

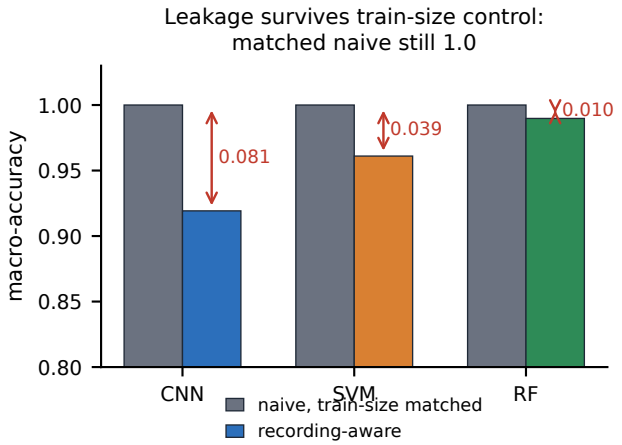

Figure 6. Measured confound control. With the naive training set subsampled to the recording-aware per-class size, naive macro-accuracy remains a perfect 1.000 for every model (grey bars), while the recording-aware estimate (coloured) sits below it; the annotated gap is the leakage that survives the training-size match, largest for the CNN.

and -0.029 for the CNN. Only the support-vector machine showed a positive drop of 0.069, and that drop is asymmetric: it came almost entirely from one pairing, the no-load target $\{1, 2, 3\}$ tested on $\{0\}$, where the SVM reproducibly fell to about 0.786 while the other direction stayed near a perfect score. We report this asymmetry openly rather than averaging it away, since a single direction of transfer can be unrepresentatively easy or hard.

Table 4. Cross-load generalization: recording-aware macro-accuracy versus cross-load macro-accuracy per model, with the cross-load drop (recording-aware minus cross-load), all mean±SD. Cross-load is averaged over the two disjoint load pairings and is the noisiest protocol; only the SVM shows a positive drop, driven by the asymmetric no-load target.

Model	Recording-aware	Cross-load	Cross-load drop
CNN	0.919±0.079	0.948±0.033	−0.029
SVM	0.961±0.027	0.893±0.107	0.069
RF	0.990±0.010	1.000±0.000	−0.010

5. Analysis

5.1. Reading the Leakage Gap

The central object of this study is a gap, not a peak, and the measured gap is real but modest on this setup. Under the naive split a model can succeed by matching a test window to its near-duplicate neighbour in training, an ability that vanishes under the recording-aware split because no recording contributes windows to both sides. The drop between the two is a direct estimate of how much apparent accuracy rests on segment-overlap leakage rather than on fault recognition, and Table 3 shows it survives the obvious confound: with the naive training pool subsampled to the recording-aware per-class size, naive macro-accuracy is still perfect for all three models, so the gap of 0.081 for the CNN, 0.039 for the SVM, and 0.010 for the random forest is attributable to the boundary and not to the larger training set the naive protocol enjoys. The magnitude, though, is small rather than dramatic, and the reason is structural to this benchmark. Collapsing the diameter-and-load taxonomy to four health states leaves several sibling recordings of every fault type on the training side, so a held-out recording is recognized from its same-class siblings even when its own near-duplicate windows are removed; the within-recording near-duplication that the recording-aware split deletes is therefore only part of the redundancy, and a residual between-recording similarity remains that no recording-grouping can remove. This is exactly the boundary the proposal anticipated, that recording-aware splitting cannot erase genuine similarity between distinct recordings of the same fault, now confirmed as the explanation for why the honest gap is modest here; it would widen with fewer recordings per class. The reading is consistent with the broader finding that train/test contamination produces overoptimistic results across machine-learning science (Kapoor & Narayanan, 2023; Roberts et al., 2021), and it refines the CWRU-specific observation that conventional splits leak bearing information (Hendriks et al., 2022): removing within-recording near-duplication shifts the estimate honestly downward, but the easy four-class task keeps

the shift small.

5.2. Deep versus Classical under an Honest Boundary

Running both families through the identical protocols answers which relies more on the removed shortcut, and the answer is the compact CNN. Figure 5 frames the contrast: under the naive split all three families saturate and are indistinguishable, so the naive protocol has no power to rank them, but under the recording-aware split they separate, and the ordering is the random forest highest, the support-vector machine next, and the compact CNN lowest. The deep model does not beat the classical baseline once the evaluation is honest; it is the weakest of the three on the recording-aware split and it also carries the largest leakage gap, far above the random forest's. This is consistent with learned filters latching onto recording-specific cues, the very idiosyncrasies a hand-crafted physical feature is indifferent to, more than the explicit descriptors do, and we report it plainly rather than implying the deep model won. The practical reading is that the honest boundary restores the discriminating power the naive split lacks, and once it does, a compact classical baseline is the stronger and steadier choice on this benchmark; the value of the deep model here is as the family most exposed to the leakage the protocol is built to measure, not as the top performer.

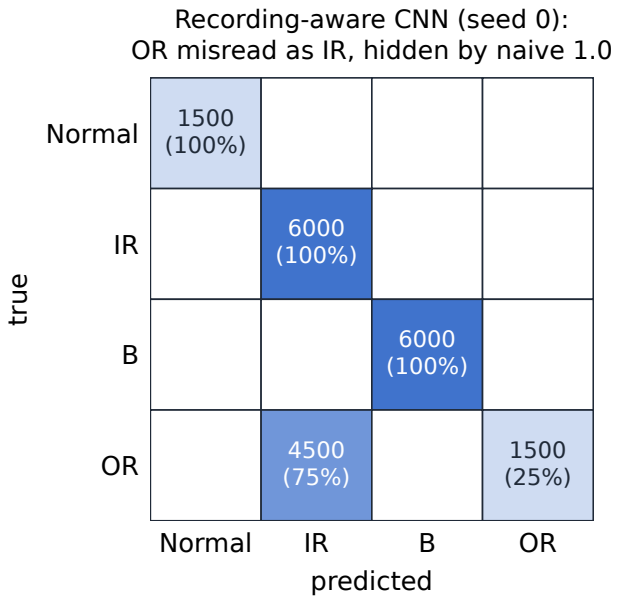

Figure 7. Measured confusion matrix of the recording-aware CNN at seed 0 (rows are true classes, cells show window counts and the within-row percentage). Naive accuracy for the same model is a perfect 1.000, yet under the honest protocol 4500 of the 6000 outer-race windows are read as inner-race, a structured per-class failure the leaky protocol masks entirely.

5.3. What Naive Accuracy Hides

The most telling consequence of leakage is not the headline number but the failure it conceals. Figure 7 shows the recording-aware CNN at one seed: outer-race faults are widely misread as inner-race, with 4500 of the 6000 outer-race windows assigned to the inner-race class and only 1500 correct, while normal, inner-race, and ball windows are classified perfectly. The same model under the naive split posts a flawless macro-accuracy, so the leaky protocol reports a perfect score on a model that, evaluated honestly, cannot tell one of the four fault types from another. This is the qualitative core of the leakage argument: the inflation is not merely a few points of accuracy but the masking of a real, structured per-class breakdown that a deployed diagnoser would suffer and a naive evaluation would never reveal.

5.4. Cross-Load Sensitivity and Failure Modes

The cross-load protocol in Table 4 probes a different failure than leakage, and it behaves per model rather than uniformly. The random forest did not degrade across loads and the CNN improved slightly, both because the cross-load setup trains on three of the four loads and the extra load diversity outweighs the shift, giving small negative drops of -0.010 and -0.029. Only the support-vector machine showed a positive drop, 0.069, and that drop is asymmetric, concentrated almost entirely in the no-load target where training on $\{1, 2, 3\}$ and testing on $\{0\}$ pulled the SVM down to about 0.786 while the opposite direction stayed near perfect. Conflating an averaged cross-load number with a per-direction one would hide this, which is why we report the pairing-level behaviour. Several failure modes follow from the design and were observed rather than assumed. First, the recording-aware split shrinks the effective number of independent units to a few recordings per condition, so the estimates are high-variance by construction, not unstable: the CNN's three recording-aware seeds were 0.81, 1.00, and 0.95, an intrinsic small-benchmark spread rather than a training failure, and it is why we report means with standard deviations and treat differences smaller than their spread cautiously. Second, the leakage caveat has a boundary of its own, the between-recording similarity discussed above, so the honest numbers are a lower-variance, less inflated estimate rather than a guaranteed upper bound on difficulty (Kapoor & Narayanan, 2023). Third, the cross-load asymmetry warns that a model can look load-robust only because the train and test loads happened to be similar, so a single pairing should never be read as general robustness; reporting both directions and their spread guards against it. Finally, the deep model's combination of the lowest honest accuracy and the largest leakage gap is itself a failure mode worth flagging: on a small benchmark, the family with the most capacity to fit recording-specific cues is the one most flattered by a leaky split and most exposed once the split is

honest.

6. Conclusion

We have reframed bearing fault diagnosis on the CWRU dataset around its evaluation protocol rather than its model. Because the conventional pipeline cuts heavily overlapping windows from a small set of long recordings and then splits them at random, near-duplicate windows straddle the train/test boundary, and the resulting near-saturated accuracy can reflect recording memorization more than fault recognition. We make the boundary the controlled variable: holding a compact 1D-CNN and a classical handcrafted-feature baseline fixed, we read both down a ladder of random, recording-aware, and cross-load protocols. The measurement confirms the leakage in direction but reframes its size: every model saturates under the naive split, the recording-aware split lowers each only modestly, and matching the naive training size leaves the gap unchanged, so the inflation is leakage rather than data volume yet is small on this easy four-class task because same-class sibling recordings remain in training. The gap is largest for the compact CNN, which does not beat the classical baseline once evaluation is honest, and the leaky protocol hides a structured outer-race failure the honest one exposes. The deliverable is an honest, lower-variance generalization estimate and a documented, modest-but-real leakage caveat with reproducible splitting code, not a state-of-the-art claim. The main limitation is a single small benchmark, whose few recordings per condition make honest estimates intrinsically variable and leave a finer between-recording similarity only partly addressed. A natural next direction is to apply the same leakage-controlled protocol to larger, independent bearing benchmarks so the measured gap can be compared across datasets and operating conditions.

References

Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., and Inman, D. J. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, 388:154–170, 2017. doi: 10.1016/j.jsv.2016.10.043.

Antoni, J. The spectral kurtosis: a useful tool for characterising non-stationary signals. *Mechanical Systems and Signal Processing*, 20(2):282–307, 2006. doi: 10.1016/j.ymssp.2004.09.001.

Antoni, J. and Randall, R. The spectral kurtosis: application to the vibratory surveillance and diagnostics of rotating machines. *Mechanical Systems and Signal Processing*, 20(2):308–331, 2006. doi: 10.1016/j.ymssp.2004.09.002.

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

Cai, J., Luo, J., Wang, S., and Yang, S. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, 2018. doi: 10.1016/j.neucom.2017.11.077.

Chen, Z. and Li, W. Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network. *IEEE Transactions on Instrumentation and Measurement*, 66(7):1693–1702, 2017. doi: 10.1109/tim.2017.2669947.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. doi: 10.1007/bf00994018.

Fawcett, T. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. doi: 10.1016/j.patrec.2005.10.010.

Glowacz, A. Fault diagnosis of single-phase induction motor based on acoustic signals. *Mechanical Systems and Signal Processing*, 117:65–80, 2019. doi: 10.1016/j.ymssp.2018.07.044.

Guo, L., Li, N., Jia, F., Lei, Y., and Lin, J. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing*, 240:98–109, 2017. doi: 10.1016/j.neucom.2017.02.045.

Guo, L., Lei, Y., Xing, S., Yan, T., and Li, N. Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data. *IEEE Transactions on Industrial Electronics*, 66(9): 7316–7325, 2019. doi: 10.1109/tie.2018.2877090.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/cvpr.2016.90.

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

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint*, 2012.

Hoang, D.-T. and Kang, H.-J. A survey on deep learning based bearing fault diagnosis. *Neurocomputing*, 335:327–335, 2019. doi: 10.1016/j.neucom.2018.06.078.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N.-C., Tung, C. C., and Liu, H. H. The

empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1971): 903–995, 1998. doi: 10.1098/rspa.1998.0193.

Ince, T., Kiranyaz, S., Eren, L., Askar, M., and Gabbouj, M. Real-time motor fault detection by 1-d convolutional neural networks. *IEEE Transactions on Industrial Electronics*, 63(11):7067–7075, 2016. doi: 10.1109/tie.2016.2582729.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint*, 2015.

Janssens, O., Slavkovikj, V., Vervisch, B., Stockman, K., Loccufier, M., Verstockt, S., de Walle, R. V., and Hoecke, S. V. Convolutional neural network based fault detection for rotating machinery. *Journal of Sound and Vibration*, 377:331–345, 2016. doi: 10.1016/j.jsv.2016.05.027.

Jena, D., Sahoo, S., and Panigrahi, S. Gear fault diagnosis using active noise cancellation and adaptive wavelet transform. *Measurement*, 47:356–372, 2014. doi: 10.1016/j.measurement.2013.09.006.

Jia, F., Lei, Y., Lin, J., Zhou, X., and Lu, N. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, 72-73:303–315, 2016. doi: 10.1016/j.ymssp.2015.10.025.

Jing, L., Zhao, M., Li, P., and Xu, X. A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox. *Measurement*, 111:1–10, 2017. doi: 10.1016/j.measurement.2017.07.017.

Kapoor, S. and Narayanan, A. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4 (9):100804, 2023. doi: 10.1016/j.patter.2023.100804.

Khan, S. and Yairi, T. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107:241–265, 2018. doi: 10.1016/j.ymssp.2017.11.024.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint*, 2014.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., and Inman, D. J. 1d convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151:107398, 2021. doi: 10.1016/j.ymssp.2020.107398.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015. doi: 10.1038/nature14539.

Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., and Siegel, D. Prognostics and health management design for rotary machinery systemsreviews, methodology and applications. *Mechanical Systems and Signal Processing*, 42 (1-2):314–334, 2014. doi: 10.1016/j.ymssp.2013.06.004.

Lei, Y., Lin, J., He, Z., and Zuo, M. J. A review on empirical mode decomposition in fault diagnosis of rotating machinery. *Mechanical Systems and Signal Processing*, 35 (1-2):108–126, 2013. doi: 10.1016/j.ymssp.2012.09.015.

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

Lessmeier, C., Kimotho, J. K., Zimmer, D., and 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. *PHM Society European Conference*, 3(1), 2016. doi: 10.36001/phme.2016.v3i1.1577.

L'Heureux, A., Grolinger, K., Elyamany, H. F., and Capretz, M. A. M. Machine learning with big data: Challenges and approaches. *IEEE Access*, 5:7776–7797, 2017. doi: 10.1109/access.2017.2696365.

Li, X., Ding, Q., and Sun, J.-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172: 1–11, 2018. doi: 10.1016/j.ress.2017.11.021.

Lu, W., Liang, B., Cheng, Y., Meng, D., Yang, J., and Zhang, T. Deep model based domain adaptation for fault diagnosis. *IEEE Transactions on Industrial Electronics*, 64(3):2296–2305, 2017. doi: 10.1109/tie.2016.2627020.

Neupane, D. and Seok, J. Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review. *IEEE Access*, 8: 93155–93178, 2020. doi: 10.1109/access.2020.2990528.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Muller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in python. *arXiv preprint*, 2012.

Randall, R. B. and Antoni, J. Rolling element bearing diagnosticsa tutorial. *Mechanical Systems and Signal Processing*, 25(2):485–520, 2011. doi: 10.1016/j.ymssp.2010.07.017.

Roberts, M., Driggs, D., Thorpe, M., Gilbey, J., Yeung, M., Ursprung, S., Aviles-Rivero, A. I., Etmann, C., McCague, C., Beer, L., Weir-McCall, J. R., Teng, Z., Gkrania-Klotsas, E., , Ruggiero, A., Korhonen, A., Jefferson, E., Ako, E., Langs, G., Gozaliasl, G., Yang, G., Prosch, H., Preller, J., Stanczuk, J., Tang, J., Hofmanninger, J., Babar, J., Sanchez, L. E., Thillai, M., Gonzalez, P. M., Teare, P., Zhu, X., Patel, M., Cafolla, C., Azadbakht, H., Jacob, J., Lowe, J., Zhang, K., Bradley, K., Wassin, M., Holzer, M., Ji, K., Ortet, M. D., Ai, T., Walton, N., Lio, P., Stranks, S., Shadbahr, T., Lin, W., Zha, Y., Niu, Z., Rudd, J. H. F., Sala, E., and Schonlieb, C.-B. Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans. *Nature Machine Intelligence*, 3(3):199–217, 2021. doi: 10.1038/s42256-021-00307-0.

Shao, S., McAleer, S., Yan, R., and Baldi, P. Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics*, 15(4):2446–2455, 2019. doi: 10.1109/tii.2018.2864759.

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

Sokolova, M. and Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009. doi: 10.1016/j.ipm.2009.03.002.

Wang, B., Lei, Y., Li, N., and Li, N. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability*, 69 (1):401–412, 2020. doi: 10.1109/tr.2018.2882682.

Wen, L., Li, X., Gao, L., and Zhang, Y. A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Transactions on Industrial Electronics*, 65(7):5990–5998, 2018. doi: 10.1109/tie.2017.2774777.

Wu, C., Jiang, P., Ding, C., Feng, F., and Chen, T. Intelligent fault diagnosis of rotating machinery based on one-dimensional convolutional neural network. *Computers in Industry*, 108:53–61, 2019. doi: 10.1016/j.compind.2018.12.001.

Yan, X. and Jia, M. A novel optimized svm classification algorithm with multi-domain feature and its application to fault diagnosis of rolling bearing. *Neurocomputing*, 313:47–64, 2018. doi: 10.1016/j.neucom.2018.05.002.

Yang, B., Lei, Y., Jia, F., and Xing, S. An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings. *Mechanical Systems and Signal Processing*, 122:692–706, 2019. doi: 10.1016/j.ymssp.2018.12.051.

Zhang, W., Peng, G., Li, C., Chen, Y., and Zhang, Z. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors*, 17(2):425, 2017. doi: 10.3390/s17020425.

Zhang, W., Li, C., Peng, G., Chen, Y., and Zhang, Z. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mechanical Systems and Signal Processing*, 100:439–453, 2018. doi: 10.1016/j.ymssp.2017.06.022.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., and Gao, R. X. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115:213–237, 2019. doi: 10.1016/j.ymssp.2018.05.050.