Deep Learning for Bearing Predictive Maintenance: A review of four different architectures

Ole Behre

March 29, 2026

Abstract

Bearing degradation accounts for approximately half of all motor failures in industrial settings, making accurate remaining useful life (RUL) prediction essential for effective maintenance planning. Deep learning has emerged as a promising data-driven approach for this task. Different architectures carry distinct trade-offs regarding feature extraction, temporal modeling, and evaluation methodology. This paper critically analyzes four deep learning approaches to bearing RUL prediction: a feedforward Deep Neural Network, a Stacked Denoising Autoencoder, a CNN-based transfer learning framework, and a 2D-LSTM fusion network. We examine their methodology, architectural design choices, and reported results. Our analysis reveals a severe data leakage issue in the DNN baseline, a disconnect between unsupervised feature learning and the actual RUL regression in the autoencoder approach, unfeasibility concerns in the CNN framework, and a gap between the recurrent architecture's sequential motivation and its snapshot-based implementation. All four approaches share a fundamental limitation: none models the continuous long-term degradation trajectory, reducing each to a snapshot-to-RUL mapping.

Contents

1 Introduction

Ball bearings are essential, highly versatile components in industry. Their applications range from vehicles and wind turbines to highly specialized and complex industrial manufacturing machinery. According to **?**, approximately half of motor failures are caused by bearing degradation. In practice, this can lead to expensive production downtimes, worker accidents and even further machinery damage through chain reactions.

The severity of this matter motivates research and application of predictive maintenance methods. A common metric for this is the remaining useful life (RUL). By being able to accurately predict when a component is likely to fail, maintenance and production outages become plannable. Historically, predictive maintenance heavily relied on model-based approaches to estimate the RUL: These approaches incorporated mathematics, physics and domain expertise to derive estimations on, for example, how long a component would last. With the increasing availability of high-resolution sensor data, especially in the context of IoT and Industry 4.0, data-driven approaches have become applicable (**?**). These leverage machine learning methods to model directly from the available data, which makes them especially interesting for more and more diverse data from complex machinery. Still, different data-driven, deep-learning architectures show different operational constraints regarding computational overhead, data scarcity, and multi-sensor integration (**?**).

To address this, this paper is guided by the following central research question: *What are the methodological strengths, limitations, and shared challenges of distinct deep learning architectures when applied to bearing RUL prediction, specifically regarding evaluation methods and fairness, feature extraction, and temporal modeling?*

To answer this question, the remainder of the paper is organized as follows: Section 2 explains the selection of the analyzed papers. Section 3 introduces the structural approaches of the four selected papers (Deep Neural Networks (DNN) (**?**), Stacked Autoencoders (SAE) (**?**), Convolutional Neural Networks (CNN) (**?**), and Deep Recurrent Neural Networks (DRNN) (**?**)) highlighting their unique architectures and the specific challenges they face. Section 4 synthesizes these findings into a comparative discussion covering evaluation comparability, shared architectural limitations, and industrial feasibility. Finally, Section 5 concludes this paper and proposes directions for future research.

2 Methodology: Scope and Paper Selection

The application of deep learning to predictive maintenance is a growing research area. A rather recent survey by (**?**) highlights the rapid expansion and structural diversity of models within this field by analyzing 106 papers. To provide a meaningful and focused analysis, our paper restricts its scope to a specific problem: the prediction of Remaining Useful Life (RUL) in rolling bearings.

We choose the following papers because they represent different domains and structural strategies:

- **Manual Feature Reliance: ?** is included as a baseline. It utilizes a standard Deep Neural Network (DNN) that still heavily depends on traditional, manual feature engineering.

- **Unsupervised Extraction: ?** represents the shift towards automated feature extraction, utilizing Stacked Autoencoders (SAE) to find patterns in raw data.
- **Data Scarcity and Computer Vision: ?** were selected to evaluate how the field handles imbalanced or missing industrial data. Their approach leverages techniques from the image processing domain, converting 1D vibration signals into 2D images to apply Convolutional Neural Networks (CNN) and transfer learning.
- **Multi-Sensor LSTM Fusion:** Finally, **?** was chosen to represent architectures built for complex environments. Their Deep Recurrent Neural Network (DRNN) approach focuses heavily on capturing time dependencies using LSTM across multiple simultaneous sensors.

All four papers use the PRONOSTIA bearing degradation dataset (**?**), which (could have) provided a common experimental basis for comparison but, as discussed in Section 4, also limits the generalizability of any conclusions drawn.

3 Surveyed Architectures & Methodologies

This section details the structural approaches, unique capabilities, and operational challenges of the four selected deep learning models. By analyzing their distinct feature extraction mechanisms and data processing requirements, we establish a foundation for evaluating their industrial feasibility.

3.1 ? | Deep Neural Network

3.1.1 Approach

? propose a fully connected feedforward neural network (FNN) for RUL prediction. As input, they rely on manual, domain-specific feature engineering. From vibration signals they derive 9 features in total: Three time-domain features (root mean square (RMS), crest factor (CF), and kurtosis) and six frequency-domain features achieved by the novel Frequency Spectrum Partition Summation (FSPS). The latter uses a Fourier transform to split the vibration spectrum into $K = 6$ frequency bands. As the used PRONOSTIA dataset (**?**) provides vertical and horizontal vibration sensor data, this leads to a total of 18 features.

These features are fed into a standard DNN consisting of eight hidden layers (300, 200, 150, 100, 80, 50, 30, 1). The input is normalized to [0,1] via Min-Max scaling, and the output layer uses a sigmoid activation function whose value is later mapped back to the RUL range. Hidden layers use ReLU activations. Dropout is employed for regularization. The network is trained using mean squared error as optimization objective and RMSprop as an optimizer.

The core idea of the paper is "multi-bearing collaborative prediction". Instead of training and testing on a single bearing, data from four bearings under the same operating conditions are pooled. The authors claim that this "introduces more challenging issues". A percentage of individual data points is randomly sampled as the test set, while the remainder serves as training data. Six train-test ratios are evaluated (Test: 5%, 10%, 15%, 20%, 25%, 30%) and for evaluation the performance in all scenarios is averaged.

3.1.2 Results

The DNN approach is compared against five other machine learning models: Gradient Boosted Decision Trees, Support Vector Machine, BP Neural Network [1], Gaussian Regression, and Bayesian Ridge Regression. They report a normalized MAE of 0.0270 and RMSE of 0.0414 averaged across all train-test splits, both better than all baselines. The Bayesian Ridge method performs worst with a MAE of 0.1399 and a RSME of 0.1710. The authors provide MAE and RMSE curves across different training data percentages but do not provide a detailed results table, making precise per-bearing analysis impossible from the reported data.

Notably, they do not include any other architectures in their comparison. Their method is only benchmarked against other classic ML methods with all using the same features as input.

3.1.3 Discussion

The most critical issue with this paper is its data splitting strategy, which introduces severe data leakage. The degradation of a bearing is a continuous process. The authors randomly distribute individual datapoints across all pooled degradation time series into their training and test splits. This way, it is inevitable that during testing, the model has already seen the test datapoint's direct neighbourhood. This leads to the model not being able to predict based on the features but to interpolate based on remembered training datapoints. In practice, it is doubtful that this method is able to successfully predict the RUL of a unseen bearing. For a fair evaluation, the authors would have to use a leave-one-out method were they train on three bearings and test on the fourth.

Aside from this, the architecture itself is limited. The model treats each input sample consisting of a very short timespan ($0.1s$) as an individual sample and has no temporal awareness. Looking at the features, the "total run time" is not encoded in any way. This seems unintuitive in this domain as the current health of a bearing should highly depend on its history.

Overall this paper is better seen as a baseline for the subsequent sections. It contains a severe experimental error and lacking comparison to other architectures. We still acknowledge the novel FSPS feature as a possibly reasonable contribution, even though no ablation study was conducted (which would have been questionable, again, due to the setup).

3.2 ? | Stacked Autoencoders

3.2.1 Approach

? propose a two-stage architecture that separates health stage classification from RUL regression. The motivation is that for most of a bearing's lifetime, the vibration patterns remain largely stable. Only in the final phases the vibration pattern starts to deviate. This can be seen in Figure **??**: The first half only shows insignificant looking vibrations, while there are clear deviations leading up to complete failure. When trying to train a single model on the whole lifespan, the model needs to fit both the long and steady state and the rapid bearing

[1] The authors do not cite any literature on "BP Neural Networks" nor do they explain it further. We assume they mean a shallow neural network trained with backpropagation.

decline at the end. For this reason, **?** try to separate the lifetime of a bearing in "health stages" and train specific predictors for each.

The first stage of the architecture addresses feature extraction and health stage classification. A Stacked Denoising Autoencoder (SDA) is used to initialize a Deep Neural Network which is later used for classification. First, the SDA is trained in an unsupervised manner: noise is injected into raw vibration signals, and the network learns to compress and reconstruct the clean input. The noise prevents the autoencoder from learning just the identity of the input and aims to capture more general and more robust features. Second, the reconstruction layers (the decoder) are discarded, and the pre-trained encoder weights are used to initialize a standard DNN. Third, a softmax classification layer is appended, and the full network is fine-tuned with labeled health stage data via backpropagation. The number of health stages is determined through grid search, landing on three stages with an uneven temporal division ([0, 0.5], [0.5, 0.75], [0.75, 1][2]). The used formula is chosen to reflect the accelerating nature of degradation.

The second stage handles RUL regression. For each health stage, a separate shallow ANN with two hidden layers (six and four neurons respectively) is trained. The inputs to these ANNs are not the latent features learned by the SDA, but manually, similar as in the previous section, engineered statistical features: RMS, Kurtosis, and CF at the current time step t and the previous step $t-1$. The inclusion of lagged features provides a simple sense of the degradation slope. The final RUL estimate is computed by weighting the output of each stage-specific ANN by the classification probability from the first stage.

Figure **??** from the original paper illustrates this pipeline.

3.2.2 Results

Xia et al. validate their approach on the PRONOSTIA dataset as well using 14 run-to-failure trajectories for training and three for testing, one per operating condition [3]. Ten trials are conducted to reduce the effect of randomness. Health stage classification achieves approximately 76.5% accuracy for Condition 1 and 73.3% for Condition 2, but drops to 54.4% for Condition 3.

For RUL prediction, the authors report average prediction errors of 7.53% for Condition 1, 7.61% for Condition 2, and 13.73% for Condition 3. Performance in the critical final phase is better for the first two and reported separately. In the last 10% of the degradation the model achieves errors of 1.05% for Condition 1 and 6.53% for Condition 2, but 19.68% for Condition 3.

The authors also compare their two-stage method against a single ANN fitted to the entire degradation process, showing that the two-stage approach produces markedly better predictions in the later half of the lifespan.

They explain poor result for Condition 3 with the scarcity of training data: only two run-to-failure trajectories exist for that operating condition (opposed to 6 each for 1 & 2).

[2] When strictly using their formula, the last interval ends at 0.875. We must assume they forgot to add that they fixed the end of the last interval to 1, mainly because graphs suggest it. If not, their experiment would be fundamentally flawed.

[3] Operating conditions in PRONOSTIA (**?**): Condition 1 (1800rpm, 4000N, 7 Runs), Condition 2 (1650rpm, 4200N, 7 Runs), Condition 3 (1500rpm, 5000N, 3 Runs)

Figure 1: Bearing1_1: (a) Actual waveform of first sample; (b) DI of first sample; (c) Actual waveform of last sample; (d) DI of last sample. Note the y-axis. (**?**)

3.2.3 Discussion

The two-stage design is well-motivated. By first identifying the onset of degradation and then applying a stage-specific regression model, the approach avoids forcing a single model to fit all. The strong performance in the final 10% of the lifespan is practically significant, as maintenance decisions are typically made during this phase (**?**). Still, looking at the accuracy graphs in their paper, performance for the first half could be considered non-satisfactory. Furthermore, the use of leave-one-out evaluation, where the test bearing is a complete unseen trajectory, avoids the data leakage problem identified in **?**.

However, the architecture suffers from a disconnect between its two stages. The first stage employs an unsupervised deep learning pipeline specifically designed to extract noise-resilient representations from raw vibration data. Yet the second stage, the actual RUL predictor, ignores these learned features entirely. Instead, it falls back on the same manually engineered statistical features (RMS, kurtosis, CF) that a conventional approach (**?**) would use. The SDA's representational power is thus only leveraged for an early classification step, not for the actual prediction. This blocks the advantage of deep learning and its end-to-end automated feature extraction. The authors argue that by using these features the model becomes more efficient at runtime and more adaptable.

By this logic the question arises if an SDA is even necessary at all. A simpler threshold-based health stage detection (such as the Wilson Amplitude approach used by **?** discussed in Section 3.4) followed by the same ANNs might achieve comparable results with significantly less architectural complexity and computation.

Finally, the two-stage pipeline introduces a risk of error propagation. Any misclassification in the first stage routes the sample to the wrong regression model in the second. With classification accuracy around 54% for Condition 3, nearly half of all samples are processed by an incorrect RUL predictor. The weighting of the ANNs by the classification confidence mitigates this to some degree, but does not eliminate the problem. The significantly improving accuracy in the top 10% leads to the assumption that especially the early ANNs are imprecise. Unfortunately, the paper does not analyze how classification errors correlate with RUL prediction errors, which would have been valuable.

3.3 ? | Convolutional Neural Networks (CNN)

3.3.1 Approach

Behera and Misra (**?**) reframe bearing RUL prediction as a computer vision problem to benefit from Transfer Learning (TL). Rather than processing one-dimensional vibration signals directly, they convert them into two-dimensional degradation images (DIs) using Markov Transition Fields (MTFs) (**?**). This decision is motivated by MTFs being able to produce a matrix that preserves

temporal structure as spatial texture[4]. Each 0.1-second vibration sample (2560 data points at 25.6 kHz) is transformed into a 2560×2560 image, then resized to 224×224 to match the input dimensions of standard pre-trained Convolutional Neural Network (CNN) architectures. Figure 1 shows what the transformed waves DIs look like.

The core of the approach is a "multi-model data-fusion deep transfer learning framework" (MMF-DTL). Three pre-trained CNN architectures (DenseNet201, VGG16, and ResNet50), all originally trained on ImageNet for object classification, are deployed in parallel. The number of layers fine-tuned for our bearing domain vary (20 for DenseNet201, 20 for ResNet50 and 10 for VGG16). As it is common in TL, earlier layers retain their original weights. The outputs of the three models are passed through individual dense layers, and then summed up in the "weighted feature fusion layer", where hyperparameters $\lambda_{\text{DenseNet201}}$, λ_{VGG16}, and $\lambda_{\text{ResNet50}}$ weigh each model's contribution. This fused representation is fed through a DNN (1200, 600, 300, 120 neurons), before a final regression layer outputs the predicted RUL.

The authors use only bearings from Operating Condition 1 of the PRONOSTIA dataset : Bearing1_1 (2803 samples) and Bearing1_2 (871 samples) for training, with the remaining five bearings under the same condition used for testing. Only horizontal vibration signals are considered. This follows the actual procedure for the IEEE PHM 2012 Prognostic Challenge (**?**). While the other reviewed architectures only used the full test dataset (which was released after the challenge), **?** predicted the RUL from a specific cutoff point in the test data. To prevent overfitting, they used standard image data augmentation techniques: Rotation, flipping, shearing and shifting.

Hyperparameters are tuned via manual search and compared against Bayesian optimization, random search, and Hyperband, with manual tuning yielding the best results.

3.3.2 Results

Bearing	Actual RUL	Predicted RUL	Error (%)
Bearing1_3	5730	5048	11.90
Bearing1_4	2900	1382	52.34
Bearing1_5	1610	1548	3.85
Bearing1_6	1460	1171	19.79
Bearing1_7	7570	6084	19.63
Score	0.542		
AER	21.5%		
MAE	807.4		

Table 1: Bearing RUL prediction results of the MMF-DTL framework (**?**).

The model is evaluated on three metrics from the IEEE PHM 2012 Challenge: an asymmetric Score function that penalizes late prediction more heavily

[4]Getting into exact workings of MFTs was beyond the scope for this submission. **?** back this claim by citing **?**

than ahead-of-time ones, the Average Error Rate (AER), and the Mean Absolute Error (MAE). Since the challenge format requires a single RUL prediction per test bearing from a predefined cutoff point, each test bearing yields one predicted value compared against the known ground truth. All metrics are only computed across the single predictions. Table 1 shows the per-bearing results.

The model indicates a clear tendency toward conservative predictions, which also helps with the highest Score of 0.542. The authors compare against both signal-to-image approaches and raw-signal approaches. Their method achieves improvements of approximately 50% on AER and 26% on MAE over the best other signal-to-image competitor (MSCNN (**?**)), and approximately 12.6% on AER over the best raw-signal method (EED (**?**)).

3.3.3 Discussion

The use of transfer learning is well-motivated in the context of data scarcity. With only two training trajectories available under Condition 1, training deep CNNs from scratch would almost certainly lead to overfitting. Leveraging ImageNet pre-trained weights provides a meaningful initialization, and the extensive ablation study comparing individual models, pairwise combinations, fusion strategies, and hyperparameter tuning method demonstrates a level of effort that is absent in the other reviewed papers.

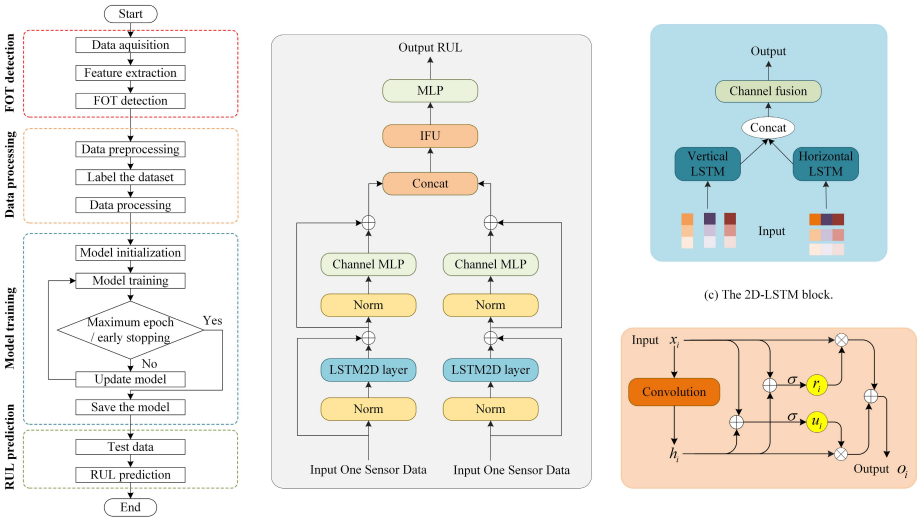
The evaluation methodology is different, as the model is tested on five entirely unseen bearing trajectories, avoiding any form of data leakage. However, it is worth noting that the challenge format evaluates only a single prediction per test bearing. With only five test bearings, the aggregated metrics are sensitive to individual outliers. This is visible in the results: the 52.34% error on Bearing1_4 compared to 3.85% on Bearing1_5 suggests that the model struggles to generalize across different trajectories within the same operating condition. A continuous evaluation across the full degradation trajectory, as performed by **?**, would provide a more robust assessment of prediction quality.

The comparison between fine-tuned and non-fine-tuned (base) models in the ablation study is a central element of their analysis, yet the results are not surprising. The base models retain their original ImageNet weights, trained to distinguish between 1000 categories of objects. Intuitively there is no reason why such a model would produce meaningful RUL regression values when presented with Markov Transition Field images of bearing vibrations.

The paper also lacks any discussion of computational feasibility. The MMF-DTL framework runs three deep CNNs (DenseNet201, VGG16, ResNet50) in parallel to do a single point prediction. Training times, inference latency, and hardware requirements are not reported. In a real application scenario, this could make this method totally unfeasible.

Furthermore, there is no explanation about how the image is downscaled to fit the CNN input, nor does the ablation study explore if the data augmentation has any benefit.

Finally, the hyperparameter tuning raises concerns. Manual tuning outperformed all three automated methods (Bayesian optimization, random search, Hyperband). The search space for the fusion weights spans ranges up to 3300, and the reported optimal values appear highly specific. Reported suggested hyperparameters seem far off, leading to the assumption that it was badly calibrated.



(a) The flowchart of proposed method. (b) The architecture of proposed model. (c) The 2D-LSTM block. (d) The information fusion unit.

Figure 2: Overall architecture of our proposed method and its components. (a) Flowchart of the proposed method. (b) Architecture of the proposed model. (c) 2D-LSTM block. (d) IFU.

3.4 ? | Deep Recurrent Neural Networks (DRNN)

3.4.1 Approach

? propose a two-stage prediction method built around what they term a "2-D LSTM fusion network". The method explicitly targets two limitations observed in earlier work: detecting when the bearing degradation process starts (here called Fault Occurrence Time (FOT)), and the difficulty of fusing information from multiple vibration sensors. The full architecture can be seen in Figure 2.

The first stage addresses FOT detection. Rather than predicting RUL across the entire bearing lifespan, the authors first identify when degradation begins. They extract the Wilson[5] Amplitude (WAMP) feature from the raw vibration signal, apply a Kalman filter for smoothing, and then compute the gradient of the WAMP values within the time window. When this gradient exceeds a predefined threshold, the bearing is considered to have entered its degradation phase. Only degradation-phase data is passed to the prediction model. The authors demonstrate that WAMP is more sensitive to the onset of degradation than commonly used features like RMS.

The second stage performs RUL prediction using a multi-subnetwork architecture. Each vibration sensor (e.g., horizontal and vertical) is processed through its own subnetwork. Before entering the model, each sensor's raw 1-D signal is transformed into a 2-D time-frequency representation (TFR) via wavelet transform and resized to 128×128. Each subnetwork then processes its TFR through a 2D-LSTM block. Within each block, a vertical LSTM and a horizontal LSTM operate in parallel on the two axes of the TFR: one sweeping across the time axis and the other across the frequency axis. Their outputs are concatenated and projected through a fully connected layer. Each block also

[5] In the literature, more often called Willison Amplitude (**?**)

includes a MLP unit, with both modules employing residual connections and layer normalization.

The outputs of the two subnetworks are concatenated and fed into an Information Fusion Unit (IFU). The IFU applies a convolutional layer to extract cross-channel spatial correlations, then uses a gating mechanism (reset and update gates) to produce a weighted combination of the convolution output and the original input. A final linear regression layer with sigmoid activation produces the predicted RUL, naturally normalized to $[0, 1]$.

3.4.2 Results

The authors validate their approach on the PRONOSTIA (**?**) dataset and the XJTU-SY (**?**) bearing dataset. On PRONOSTIA, they use six bearings for training (two per operating condition) and the remaining eleven for testing. On XJTU-SY, one bearing per condition is reserved for testing.

? report "lowest prediction errors in all conditions compared to other models mentioned". Their method achieves the lowest RMSE across all eleven PRONOSTIA test bearings compared to five other architectures. On the XJTU-SY dataset, the model outperforms all baselines on all three test bearings. Notably, for the PRONOSTIA dataset, for three bearings (1_3, 1_6, 1_7) Cap-LSTM (**?**) achieves better MAPE scores, indicating competitive and even better relative accuracy in a few cases.

At the end, an ablation study systematically evaluates each component. Replacing the 2D-LSTM with a standard LSTM degrades performance in 9 of 11 test cases. Adding multi-sensor input improves accuracy in 8 of 11 cases. The IFU consistently improves all configurations.

3.4.3 Discussion

The FOT detection via WAMP gradient is a practical contribution. It solves the same problem that motivated **?**'s health stage classification, but with a simple heuristic rather than a deep classifier.

The IFU is the most architecturally novel component. Approaches such as **?** treat horizontal and vertical vibration sensor readings as independent, or just pick one **?**. The IFU's convolutional mixing followed by gated filtering provides a learnable mechanism for capturing this coupling, and the ablation results confirm its benefit.

While the 2D-LSTM effectively captures patterns along both axes of the time-frequency input, each input is a single 128×128 representation derived from one short vibration sample. The LSTM's hidden state operates within this window but is not carried across successive samples, meaning the model has no mechanism for learning macro degradation trends over hours or days. Like the other approaches reviewed in this paper, it is fundamentally a snapshot-to-RUL mapping.

Finally, the computational cost of the architecture is also not addressed. The multi-subnetwork design multiplies parameters with each additional sensor, and the sequential nature of LSTM processing limits parallelization which is a potential concern for industrial deployment relative to CNN-based alternatives.

4 Discussion

4.1 Evaluation & Comparability

All four approaches use the PRONOSTIA dataset (**?**), yet their results are not directly comparable. Each study selects different subsets of bearings for training and testing, applies different evaluation metrics, and defines the end-of-life threshold differently. **?** report normalized MAE and RMSE on randomly split data points. **?** report life percentage prediction error on whole trajectories. **?** report AER and MAE for one specific time stamp per test bearing. **?** report RMSE and MAPE on normalized RUL values. The absence of a standardized evaluation protocol across the field makes quantitative architectural comparison impossible from published results alone.

Beyond the metrics, the evaluation rigor varies substantially. As discussed in Section 3, the random point-splitting strategy in the DNN approach introduces data leakage that likely inflates their reported accuracy. The remaining three papers use trajectory-level separation, which is methodologically sound. However, the PRONOSTIA dataset itself is a limiting factor with only 17 run-to-failure trajectories across three operating conditions. **?** results illustrate this clearly — with only two training trajectories, classification accuracy drops to 54% and prediction error nearly doubles.

We consider this dataset to be of fundamental importance in the field. However, it is now over 13 years old, contains only a limited amount of data, and, as demonstrated in this paper, has not established as an comparable benchmark across research.

4.2 Temporal Nature of Degradation

Despite representing four distinct architectures, all four investigated approaches share a fundamental constraint. None models the continuous, long-term degradation trajectory of a bearing. Each processes isolated short-duration snapshots (0.1 seconds for PRONOSTIA) and maps them independently to a RUL estimate. The DNN has no temporal mechanism at all. The SA incorporates a single lag at $t-1$ providing only a local gradient. The CNN encodes only snapshots into static images. The DRNN , despite being architecturally designed for sequential data, resets its hidden state between snapshots.

This means that all four models are fundamentally performing the same task: inferring the remaining useful life from the localized vibration pattern of a single moment. The architectural differences determine how each model extracts features from that snapshot, but none possesses awareness of where that snapshot sits in the entire degradation history beyond what the features encode. Addressing this limitation is arguably constrained primarily by the computational cost of processing thousands of high-frequency vibration recordings sequentially.

4.3 Feature Engineering vs. Deep Learning

The four papers illustrate a progression from manual to automated feature extraction, though this progression is not as clean as their architectures might suggest. The DNN relies entirely on hand-crafted features. The SA approach uses unsupervised deep learning for classification but then goes back to manual

features for the actual RUL regression. The CNN approach achieves fully automated feature extraction through transfer learning, but requires an intermediate signal-to-image transformation whose computational feasibility and necessity is questionable. The DRNN also automates feature extraction via convolution and learned representations, though the FOT detection still relies on a selected threshold.

In practice, none of the four approaches achieves true end-to-end learning from raw vibration signals to RUL prediction without inductive bias at some stage of the pipeline. We argue that for bearing RUL prediction on small datasets, purely automated feature extraction may not yet be sufficient, and hybrid approaches that combine domain knowledge with learned representations remain the most practical path. Still, transfer learning showed promising results and domain specific data augmentation techniques could be explored.

4.4 Industrial Feasibility

None of the reviewed approaches are tested under variable operating conditions in a realistic sense, mainly because the shared dataset provided only three static conditions (fixed RPM and load). In a live manufacturing environment, machines routinely shift between loads, which fundamentally alters the vibration signature. Whether any of the reviewed architectures would generalize to such dynamic settings is unknown, though in a side experiment (**?**) show how their architecture supports an adaptive updating mechanism. Still, computational feasibility is rather discussed as an excuse for shallow ANNs for prediction than in actual real-world scenarios. The seen time windows in this paper are of 10 milliseconds and contain thousands of data points. This either calls for extreme efficiency or the ability to quickly filter for relevant data points in longer timeframes.

Furthermore, all four approaches treat RUL as a deterministic point estimate. For maintenance scheduling, a prediction of "807 time steps remaining" (as produced by **?**) is less actionable than a probabilistic forecast with confidence intervals. Only **?** provide any uncertainty information through the classification probabilities that weight their stage-specific predictors, but this is a byproduct of the architecture rather than an explicit uncertainty quantification.

On a final note, none of the architectures discuss the factor of explainability. In a high-stakes environment like a factory, decision makers want to know how a model came to a decision before e.g. halting production for maintenance. Here simple, well engineered features with clear assigned weights and threshold still have the edge over deep learning architectures.

5 Conclusion

This paper analyzed four deep learning architectures for bearing RUL prediction: a standard DNN, a Stacked Denoising Autoencoder, a CNN with transfer learning, and a 2D-LSTM fusion network. We specifically focused on their operational trade-offs regarding feature extraction, temporal modeling, evaluation rigor, and industrial feasibility.

Our analysis reveals that architectural sophistication does not straightforwardly translate to prognostic capability. The simplest approach (**?**) achieves

seemingly strong results that are largely attributable to data leakage rather than genuine prediction. The most complex approach (**?**) delivers the most consistent performance across the dataset, but shares the same fundamental limitation as all reviewed methods: no architecture models the continuous degradation trajectory, reducing each to a snapshot-to-RUL mapping regardless of its internal complexity.

We further identified a recurring disconnect between deep learning's promise of automated feature extraction and its practical implementation. Of the four approaches, none achieves true end-to-end learning without manual intervention, whether through hand-crafted features (**?**), a feature disconnect between stages (**?**), questionable image transformations (**?**), or previously selected detection thresholds (**?**).

Beyond these architectural concerns, none of the reviewed approaches address key requirements for industrial deployment. All four treat RUL as a deterministic point estimate, lacking the probabilistic forecasts with confidence intervals that maintenance scheduling might demand. None are validated under variable operating conditions, and none discuss explainability which is a critical factor when models inform decisions such as halting production.

For future work, we see two priorities. First, architectures that maintain state across entire degradation trajectories rather than isolated snapshots would represent a genuine advance over the current paradigm. Second, the field requires standardized evaluation protocols on larger, more diverse datasets. The community's reliance on the 17-trajectory PRONOSTIA dataset limits the generalizability of any architectural comparison, including this one.