Contrastive, Autoencoding, and Variational Representations for Telemetry-Driven RUL Prediction

Mahmoud Rahat[1]

[1] *Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Sweden*
mahmoud.rahat@hh.se

ABSTRACT

Predictive Maintenance in safety-critical industries relies on accurate Remaining Useful Life (RUL) estimation from multivariate telemetry. Still real-world datasets are often dominated by censored observations and frequently lack explicit failure annotations. These constraints limit the effectiveness of purely supervised learning and motivate the need for approaches that can leverage unlabeled data. This paper presents a Pseudo RUL Guided Semi Supervised Learning framework that combines unsupervised representation learning with physics and statistics based soft failure indicators to enable robust RUL prediction under scarce failure labels. Compact latent representations are learned from censored telemetry using three encoder families i.e. autoencoders, variational autoencoders, and contrastive learning. The learned representations are subsequently used as inputs to a lightweight regression model trained on the available labeled samples. In scenarios where no failures are recorded, soft-failure transitions are used to construct pseudo-RUL targets, allowing training to proceed even in fully censored settings. Experiments on three diverse multivariate time-series datasets demonstrate that the learned representations consistently reduce prediction error relative to raw features while also reducing model size.

1. INTRODUCTION

The increasing availability of large-scale IoT and telemetry streams from industrial equipment has enabled data-driven approaches for Prognostics and Health Management (PHM). A central task in this domain is the prediction of future failures through Remaining Useful Life (RUL) estimation. In a typical setting, each device generates multivariate time-series observations over its operational lifetime, and RUL labels are constructed from the time remaining until the next failure event. However, in real-world fleets, the vast majority of devices never experience a recorded failure within the study period. As a consequence, only a small fraction of the observations can be assigned a true RUL value, while the rest must be treated as censored.

To address this imbalance, I explore whether the abundant *unlabeled* samples can be exploited through unsupervised learning. The central idea of this paper is to use the telemetry from non-failing devices to learn a compact latent representation of system behaviour. I investigate three widely used manifold-learning methods for multivariate time series: (i) Autoencoders (AE), (ii) Variational Autoencoders (VAE), and (iii) Contrastive Learning. After pretraining, the decoder portion of each model is discarded, and the encoder is augmented with a multilayer perceptron (MLP) RUL prediction head. The RUL head is then fine-tuned using only the RUL-labeled samples while keeping the encoder layers frozen.

A second challenge in operational fleets arises when explicit maintenance or failure records are missing entirely. This situation is common for several reasons: companies may not release detailed failure logs due to confidentiality concerns; equipment may be part of a young fleet with few or no recorded failures; or the underlying hardware may be highly reliable, resulting in extremely rare true failure events. In such settings, direct RUL labeling is impossible. To overcome this, I investigate whether *soft-failure points* i.e. early indicators of degraded or abnormal performance can be used to construct *pseudo-RUL* targets. These soft-failure indicators are derived from domain informed degradation signals and represent transitions from healthy to unhealthy operation rather than hard failures.

This work is driven by the following research questions:

1. Can unsupervised representation learning on censored telemetry improve supervised RUL prediction performance compared to using only failure-labeled samples?
2. Among AE, VAE, and Contrastive Learning, which representation learning method produces the most informative manifold for downstream RUL estimation?
3. In the absence of explicit failure records, can we define pseudo-RUL values based on soft-failure indicators reliably enough to support RUL prediction?

By addressing these questions, we aim to bridge the gap between real-world telemetry which is often rich in data but poor in labels and the supervised learning requirements of modern RUL prediction pipelines.

2. RELATED WORK

Real-world predictive maintenance datasets present three compounding challenges that standard supervised approaches cannot address in isolation: the scarcity of failure labels, the dominance of censored observations, and the absence of explicit failure records in some operational fleets. This section reviews prior work along three axes that directly motivate our framework: self-supervised and contrastive representation learning for time series, semi-supervised and label-efficient RUL estimation, and autoencoder-based latent representations in prognostics. Each body of work addresses part of the challenge; our framework integrates ideas from all three.

These challenges are compounded by the heterogeneity of industrial signals, which require careful adaptation before standard deep learning architectures can be applied effectively (Revanur et al., 2020; Kharazian et al., 2025). At the same time, temporal dependencies necessitate architectures such as LSTMs, which capture degradation trajectories but remain sensitive to distribution drift (Rahat et al., 2022). Beyond single-architecture solutions, ensemble strategies that combine heterogeneous base learners have also been proposed to preserve fault-detection robustness under evolving operating conditions and concept drift (Altarabichi et al., 2020). When viewed together with broad surveys of deep learning architectures for predictive maintenance, which highlight the specific strengths and weaknesses of DNNs, CNNs, SAEs, DBNs, and DRNNs, it becomes clear that no single architecture is sufficient for the multifaceted demands of industrial prognostics (Li et al., 2024).

Large-scale real-world datasets illustrate these challenges directly. The Scania Component X dataset, for example, contains high censoring rate, irregular temporal sampling, substantial cross-vehicle heterogeneity, and strong operational drift, embodying nearly all obstacles identified across the literature (Kharazian et al., 2025). Classical survival models struggle because censoring is extreme and event labels are sparse; regression models fall short because they ignore available censored information; and deep learning architectures require domain adaptation, dimensionality reduction, and censor aware learning strategies to remain stable. Addressing the mentioned gaps is essential for building reliable, real-world RUL prediction systems and forms the core motivation for developing unified survival-deep learning frameworks.

2.1. Self-Supervised and Contrastive Representation Learning for Time Series

Self-supervised representation learning has emerged as a powerful paradigm for time-series data, particularly in settings where labeled samples are scarce. Contrastive methods such as TS-TCC (Eldele et al., 2021) learn temporal representations by maximizing agreement between augmented views of the same time step across temporal and contextual dimensions. Similarly, Ts2Vec (Yue et al., 2022) proposes a universal framework for learning contextual representations at arbitrary semantic levels via hierarchical contrastive loss. Transformer-based approaches have also been applied to unsupervised multivariate time-series representation learning, demonstrating strong downstream performance on classification and regression tasks (Zerveas et al., 2021). More recently, temporal neighborhood coding (Tonekaboni et al., 2021) exploits the local smoothness assumption in physiological and sensor signals to define positive pairs for contrastive objectives without requiring data augmentation. These works collectively establish that self-supervised pretraining on unlabeled time-series data can substantially improve downstream task performance. However, they are primarily validated on classification benchmarks with clean labels and do not address the censored, regression-oriented setting of RUL estimation.

2.2. Semi-Supervised and Label-Efficient RUL Estimation

Building on the representation learning foundations reviewed above, a growing body of work has applied semi-supervised principles specifically to the RUL estimation problem. LSTM-based encoder-decoders trained on normal operational data were among the earliest demonstrations that reconstruction error from unlabeled sequences can proxy health state (Malhotra et al., 2016). Generative adversarial networks have since been adapted to RUL directly, with (He et al., 2022) proposing a semi-supervised GAN that jointly exploits failure histories as labeled data and suspension histories as unlabeled data. (Krokotsch et al., 2021) showed that self-supervised pretraining on unlabeled run data consistently outperforms competing SSL approaches on NASA C-MAPSS, particularly when data near failure is withheld. Despite this progress, most existing methods assume that at least some run-to-failure trajectories are available. The fully censored scenario, in which no failure labels exist and pseudo-targets must be constructed from soft-failure indicators, remains largely underexplored.

2.3. Autoencoders and Variational Methods in Prognostics

The semi-supervised approaches reviewed above rely on encoder architectures to produce compact latent representations. Here we review the autoencoder and variational autoencoder families that form the backbone of our pretraining stage. Stacked and recurrent autoencoders trained on unlabeled sensor streams

were among the first deep architectures shown to produce latent representations that implicitly encode degradation severity, with reconstruction error serving as a monotonic health indicator (Malhotra et al., 2016). Variational autoencoders (Kingma et al., 2013) extend this by imposing a structured probabilistic prior on the latent space: (Costa et al., 2021) introduced a recurrent VAE for RUL estimation whose latent trajectory provides an interpretable view of engine deterioration, while (Costa et al., 2022) showed that KL divergence regularization encourages a smooth latent manifold that correlates with degradation even without explicit RUL labels. Dynamic conditional VAEs have further been proposed for health index construction under varying operational conditions (Wei et al., 2021), and dynamical VAE variants have been applied to uncertainty-aware RUL estimation producing probabilistic rather than point predictions (Star et al., 2025). Taken together, these works motivate our choice of AE, VAE, and contrastive learning as the three encoder families to compare. Each one offers a distinct inductive bias for organizing the latent space and where previously relatively unexplored in a setting similar to our work.

3. DATASETS

In this section, we present the three datasets used in this study, each selected for its distinct characteristics and relevance to prognostics and health management. These datasets are used both in the methodology section and in the experimental evaluation.

3.1. C-MAPSS

The C-MAPSS dataset (Saxena et al., n.d.) is a widely used benchmark for aircraft engine prognostics created using NASA's Commercial Modular Aero-Propulsion System Simulation model. It provides run-to-failure multivariate time-series data generated by simulating degradation in key engine modules such as the fan, compressors, and turbines. The resulting dataset contains realistic sensor response surfaces, noise, and operational variability, making it a standard testbed for Remaining Useful Life (RUL) prediction research.

3.2. Component-X

The second dataset is called Scania Component X. The Scania Component X dataset is a real-world, multivariate time-series resource for predictive maintenance that combines (i) operational readouts (1,122,452 observations from 23,550 vehicles with 107 columns), (ii) time to event (TTE) / repair records that indicate whether Component X was repaired during the study period, and (iii) vehicle specifications—all anonymized and privacy-preserving.

3.3. Water-Treatment

The third dataset comes from a fleet of industrial water treatment devices. The dataset is proprietary, fully anonymized, and was provided to the researchers for a limited time. Each row corresponds to an operating interval rather than a single sensor snapshot, defined by a start timestamp and a duration measured in seconds. The dataset includes both operational-context variables describing processes such as loading, unloading, cleaning-in-place, and standby, as well as physical measurements such as temperature, pressure, and flow. In addition, timing features indicate how long the system remained in various internal states during each interval. The final dataset contains 162,706 rows and 385 columns, of which 381 are numerical. A unique device identifier (machine_id) links observations from 117 distinct units. Temporal information is provided through a time column spanning roughly three years of observations.

4. METHODOLOGY

This section describes the two-stage framework proposed in this work. We first detail the three unsupervised representation learning strategies used for pretraining, followed by the RUL labeling procedures applied to each dataset.

4.1. Representation Learning

This study investigates whether self-supervised representation learning can improve Remaining Useful Life (RUL) prediction from telemetry data. We compare three representation learning strategies .i.e. contrastive learning, autoencoders (AE), and variational autoencoders (VAE) and evaluate their effectiveness for downstream RUL regression. All representations are learned exclusively from censored (unsupervised) samples. The pretrained encoder is then used to map the supervised samples into a latent vector. Finally, a multilayer perceptron (MLP) regressor is trained and evaluated on these learned projections to assess the impact of self-supervised pretraining on RUL prediction performance. The overall framework is illustrated in Figure 1.

The downstream MLP regressor is exclusively trained using supervised samples (i.e. samples whose their end of life is observed by the end of the observation window). This means the censored samples are excluded during the regressor training. Although a lower bound on RUL can be derived for censored samples as $\mathrm{RUL}(t) \geq t_{\text{end}} - t$, we deliberately exclude these samples from the supervised regression loss for two reasons. First, treating the lower bound as a point target introduces systematic bias, as surviving assets are unlikely to fail immediately at the end of the observation window. Second, properly incorporating inequality-constrained targets requires a censoring-aware loss function, such as those used in survival analysis e.g., (Rahat et al., 2024), which is beyond the scope of this work. Instead, censored samples are fully ex-

ploited during the unsupervised pretraining stage, where the learned representations encode degradation patterns from the entire fleet including non-failing assets. Below we briefly discuss the approaches used to produce the learned representations.

4.1.1. Contrastive Representation Learning

For contrastive learning, an encoder network $f_\theta(\cdot)$ maps each input vector to a latent representation:

$$\mathbf{z}_i = f_\theta(\mathbf{x}_i). \tag{1}$$

Two stochastic views of each unlabeled sample are generated by applying additive Gaussian noise:

$$\tilde{\mathbf{x}}_i^{(1)} = \mathbf{x}_i + \boldsymbol{\epsilon}_i^{(1)}, \quad \tilde{\mathbf{x}}_i^{(2)} = \mathbf{x}_i + \boldsymbol{\epsilon}_i^{(2)}, \tag{2}$$

where $\boldsymbol{\epsilon}_i^{(1)}$ and $\boldsymbol{\epsilon}_i^{(2)}$ are Gaussian perturbations. The encoder output is passed through a projection head $g_\phi(\cdot)$, and the model is trained using the normalized temperature-scaled cross-entropy (NT-Xent) loss. This objective encourages two augmented views of the same sample to be close in latent space while pushing apart representations from different samples.

After contrastive pretraining using unlabeled data, the projection head is discarded. The pretrained encoder is then frozen (i.e., its parameters are not fine-tuned) and connected to a multilayer perceptron (MLP) regressor, which is later trained on the labeled samples for RUL prediction.

4.1.2. Autoencoder Representation Learning

In the autoencoder approach, an encoder-decoder pair is trained on unlabeled data to reconstruct the input:

$$\mathbf{z}_i = f_\theta(\mathbf{x}_i), \quad \hat{\mathbf{x}}_i = h_\psi(\mathbf{z}_i). \tag{3}$$

The model is optimized using the reconstruction loss

$$\mathcal{L}_{AE} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2. \tag{4}$$

Once pretraining is completed, the encoder is frozen and connected to an MLP regressor for supervised RUL estimation on labeled samples in the next step.

4.1.3. Variational Representation Learning

The variational autoencoder (VAE) extends the autoencoder by learning a probabilistic latent representation. For each input $\mathbf{x}_i$, the encoder outputs the mean and log-variance of a latent Gaussian distribution:

$$(\boldsymbol{\mu}_i, \log \boldsymbol{\sigma}_i^2) = f_\theta(\mathbf{x}_i). \tag{5}$$

A latent code is then sampled using the reparameterization trick:

$$\mathbf{z}_i = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{6}$$

The VAE is trained by minimizing a combination of reconstruction loss and Kullback-Leibler divergence:

$$\mathcal{L}_{VAE} = E\left[\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2\right] + D_{KL}\left(q(\mathbf{z}_i|\mathbf{x}_i) \,\|\, p(\mathbf{z}_i)\right). \tag{7}$$

After pretraining, the mean latent representation is later used as input to an MLP regressor for downstream RUL prediction.

4.2. RUL labeling

Samples get RUL labels in slightly different ways. The labeling strategy for all three datasets is summarized below.

4.2.1. Labeling C-MAPSS Dataset

Let u index engines and $t \in \{1, \ldots, T_u\}$ index cycles within engine u. For each engine u, we compute the last observed cycle

$$T_u \;=\; \max(\texttt{time\_step}_u).$$

The *true* Remaining Useful Life (RUL) at cycle t is then

$$\mathrm{RUL}_{u,t}^{\mathrm{true}} \;=\; T_u - t.$$

To enable semi-supervised training and inspired by (Rahat et al., 2023), we intentionally leave early-life samples unlabeled. Given an *unlabeled ratio* $\rho = 0.5$, we define the engine-specific cutoff

$$\mathrm{cutoff}_u \;=\; \Big\lfloor T_u \times \rho \Big\rfloor.$$

This partitions the observations of each engine u into a labeled set $\mathcal{L}_u$ and an unlabeled (censored) set $\mathcal{U}_u$:

$$\mathcal{L}_u \;=\; \big\{t \mid t > \mathrm{cutoff}_u\big\}, \qquad \mathcal{U}_u \;=\; \big\{t \mid t \le \mathrm{cutoff}_u\big\}.$$

The supervised regression loss is computed exclusively over labeled samples:

$$\mathcal{L}_{\text{sup}} \;=\; \frac{1}{|\mathcal{L}|}\sum_{u}\sum_{t\,\in\,\mathcal{L}_u} \big|\mathrm{RUL}_{u,t}^{\mathrm{true}} - \hat{y}_{u,t}\big|,$$

where $\mathcal{L} = \bigcup_u \mathcal{L}_u$ denotes the full labeled set across all engines. Samples in $\mathcal{U}_u$ are excluded from this loss and are used during the unsupervised pretraining stage, where the learned representations encode degradation patterns from the entire fleet including non-failing assets.

4.2.2. Labeling Component X Dataset

In the Scania Component X dataset, the Remaining Useful Life (RUL) is derived directly from the time-to-failure information available through the repair and failure records associated with each vehicle. Because these records explicitly in-

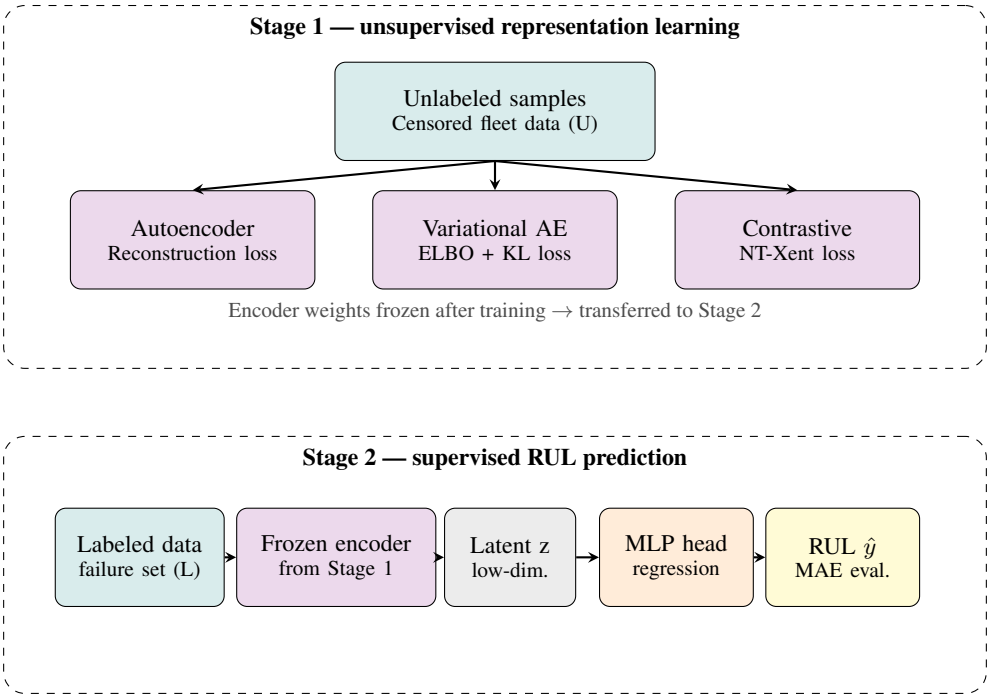

Figure 1. Overview of the proposed two-stage framework. In Stage 1, an encoder is pretrained on unlabeled censored samples using one of three unsupervised objectives: reconstruction loss (AE), ELBO with KL regularization (VAE), or NT-Xent contrastive loss. Encoder weights are frozen and transferred to Stage 2, where labeled samples are projected into a compact latent vector consumed by a lightweight MLP regressor trained to predict RUL.

dicate whether a failure occurs within the study period, the labeling procedure is analogous to the C–MAPSS formulation and is therefore straightforward.

For any observation at time t, if $\texttt{in\_study\_repair} = 1$, meaning that the component fails during the study period, the RUL corresponds to the remaining time until the next failure event. Let t_{failure} denote the timestamp of that failure. The RUL is then defined as

$$\text{RUL}(t) = t_{\text{failure}} - t, \tag{8}$$

which matches the C–MAPSS convention of measuring the remaining lifetime relative to the final observed event of the unit. Thus, each labeled sample reflects the interval between the current timestamp and the next failure (or repair) event of Component X.

If $\texttt{in\_study\_repair} = 0$, no failure is observed for that vehicle within the study window. In this case, the sample is treated as censored and added to the unlabeled set $\mathcal{U}$, excluded from the supervised regression loss and used exclusively during the unsupervised pretraining stage.

4.2.3. Labeling Water-Treatment Dataset

In the water treatment dataset, explicit maintenance, replacement, or failure records are not available. Instead, the dataset contains a set of anomaly indicators that reflect transitions from healthy to degraded operation. These anomaly indicators represent *soft failures* or *performance-based degradation events* in PHM terminology: they do not correspond to complete system failure, but rather mark the onset of persistent abnormal behavior.

A group of eleven degradation-related signals—referred to as the core degradation signals—was identified in collaboration with domain experts from the company. Each feature corresponds to a specific type of abnormal or faulty behavior. A degradation signal is considered *active* when its value is strictly greater than zero. A *degradation episode* for a given unit begins when one or more core degradation signals are active for three consecutive rows, indicating persistent abnormal behavior. This persistent-activity rule serves as the criterion for defining soft failure events in the dataset. This three-consecutive-row criterion was established in collaboration with domain experts from the company and reflects the operational definition of a persistent fault in these devices: a single active degradation signal is considered noise, whereas three consecutive activations indicate a genuine and sustained transition to degraded operation. We acknowledge that these soft-failure events are proxy signals rather than confirmed hard failures. However, we argue that this is both a practical necessity and a deliberate design choice.

In many real-world industrial settings, true failure labels are unavailable, either because failures are extremely rare, be-

cause degradation is gradual and difficult to pinpoint to a single event, or because the complexity of the degradation process makes precise failure annotation infeasible within realistic time and resource constraints. In such settings, waiting for actual breakdowns is neither practical nor safe. Instead, predicting the onset of persistent anomalous behavior — as we do here — is itself a meaningful and actionable objective for the company, providing early warning of deteriorating conditions before they escalate into catastrophic failures. We therefore argue that the ability to predict such soft-failure transitions is a valuable contribution in its own right, independent of whether these proxy events can be directly aligned with hard failure records, which in this case are unavailable. This view is consistent with recent PHM literature, which increasingly recognizes early anomaly onset detection as a meaningful prognostic target in settings where hard failure labels are unavailable or unreliable (Kamat et al., 2021). This is further reinforced by evidence that even nominally hard failure labels in industrial datasets are frequently compromised by mislabeled premature replacements (Alabdallah et al., 2023), undermining the assumption that ground truth annotations are inherently more reliable than carefully constructed proxy signals.

Once degradation episodes are identified, RUL labels are constructed in a manner consistent with the other datasets. Let t denote the current observation time for a unit, and let t_{event} denote the timestamp at which the next degradation episode *starts*. Then:

$$\mathrm{RUL}(t) = t_{\text{event}} - t,$$

which defines the time (or steps) remaining until the onset of the next persistent degradation episode. Samples that already lie inside a degradation episode are assigned

$$\mathrm{RUL}(t) = 0.$$

If no future degradation episode occurs after time t, the sample is treated as censored and added to the unlabeled set $\mathcal{U}$, excluded from the supervised regression loss and used exclusively during the unsupervised pretraining stage.

In summary, degradation episodes represent early warnings of emerging persistent faults rather than hard failures. The RUL labels capture the time remaining before such soft failures occur, allowing the model to learn to anticipate the transition from healthy to degraded operation.

5. EXPERIMENTS

For all experiments, labeled samples are split into training, validation, and test subsets using a fixed random seed for reproducibility. To prevent data leakage, the split is performed at the unit level: all observations belonging to a given unit are assigned exclusively to one fold. A fixed random seed is used to ensure that the unit-to-fold assignment is reproducible across runs. We then perform 5-fold cross-validation over units, guaranteeing that no unit appears in more than one split and eliminating any risk of temporal or cross-unit leakage. Performance is reported as mean $\pm$ standard deviation across the five folds. Unlabeled samples are used only during the self-supervised pretraining stage, whereas labeled samples are used for downstream regression training and evaluation. Model performance is reported using mean absolute error (MAE) on the held-out test set.

5.1. Baseline Models

Two baselines are considered for comparison. The first is a plain supervised MLP trained directly on labeled input features without any unsupervised pretraining. The second is a mean-value predictor that estimates the RUL of each test sample as the average RUL observed in the training set. The second baseline represents an absolute minimum level of performance, with any value below it effectively corresponding to a random guess. These baselines enable assessment of whether the learned latent representations provide gains over direct supervised learning and naive prediction.

5.2. Training Setup

The Component_X, Water_Treatment, and C_MAPSS datasets contain 107, 385, and 26 features, respectively. Both Component_X and C_MAPSS used the same multilayer perceptron architecture (input $\rightarrow$ 128 $\rightarrow$ 64 $\rightarrow$ 32 $\rightarrow$ 1) for the baseline model and the MLP heads, whereas the Water_Treatment dataset employed a wider baseline MLP architecture (input $\rightarrow$ 256 $\rightarrow$ 128 $\rightarrow$ 64 $\rightarrow$ 1) due to its substantially larger feature dimension. A detailed comparison of the number of trainable parameters across all datasets and model variants is provided in Table 1. Early stopping is employed during both pretraining and regression training to reduce overfitting and preserve the best model according to validation performance.

Let $\mathcal{L}$ denote the set of labeled samples and $\mathcal{U}$ the set of unlabeled (censored) samples. The mean absolute error (MAE) is computed only over $\mathcal{L}$:

$$\mathcal{L}_{\text{sup}} = \frac{1}{|\mathcal{L}|} \sum_{i \in \mathcal{L}} |y_i - \hat{y}_i|$$

where y_i and $\hat{y}_i$ denote the true and predicted RUL values, respectively. The final comparison is reported across contrastive learning, autoencoder, variational autoencoder, plain MLP, and mean-value prediction.

5.3. Results on Component-X Dataset

Table 2 presents the results obtained from experiments on the Component-X dataset. AE+MLP model achieved the lowest MAE among the evaluated methods. Figure 2 shows the two-dimensional t-SNE projection of the latent space learned by the best-performing model. Each point corresponds to a la-

Table 1. Number of trainable parameters for baseline and representation-based models across all datasets. Note that the parameters used in representation layers are excluded since they are kept frozen to keep the experiments fair.

Dataset	MLP	Rep+MLP	Latent Size
C-MAPSS	13,825	11,777	10
Component-X	24,961	16,897	50
Water-Treatment	139,777	54,273	50

Table 2. Final performance comparison on Component-X in terms of MAE. Values are reported as mean ± standard deviation across five folds

Method	MAE
Contrastive + MLP	36.3782 ± 0.1463
AE + MLP	$\mathbf{34.0910} \pm 0.4101$
VAE + MLP	36.7250 ± 0.3958
Standalone MLP	45.7233 ± 0.2888
MeanValuePredictor	69.0682

beled sample, and the color indicates remaining useful life, with red denoting samples close to failure and green denoting samples farther from failure. Figure 3 presents the training dynamics of the same model, including the representation-learning stage and the MLP head fine-tuning stage.

5.4. Results on C-MAPSS Dataset

Table 3 presents the results obtained from experiments on C-MAPSS Dataset. The VAE+MLP model achieved the lowest MAE among the evaluated methods. Figure 4 shows the two-dimensional t-SNE projection of the latent space learned by the best-performing model. Each point corresponds to a labeled sample, and the color indicates remaining useful life, with red denoting samples close to failure and green denoting samples farther from failure. Figure 5 presents the training dynamics of the same model, including the representation-learning stage and the MLP head fine-tuning stage.

5.5. Results on Water-Treatment Dataset

Table 4 presents the results obtained from experiments on Water-Treatment Dataset. The Contrastive+MLP model achieved the lowest MAE among the evaluated methods. Figure 6 shows the two-dimensional t-SNE projection of the latent space learned by the best-performing model. Each point corresponds to a labeled sample, and the color indicates remaining useful life, with red denoting samples close to failure and green denoting samples farther from failure. Figure 7 presents the training dynamics of the same model, including the representation learning stage and the MLP head fine-tuning stage.

5.6. Discussion

All learned representations exhibit clear and meaningful latent space structure, with samples of similar RUL values clustering in the same regions. Note that this separation was achieved while training only with unlabeled samples. Among the three methods, the representation learned by the variational autoencoder is noticeably denser and displays a smooth, gradual progression of RUL values from left to right in the latent space—a pattern that suggests a more stable and robust mapping. This behavior is consistent with the VAE's strong probabilistic foundation, which naturally encourages well-organized and continuous latent manifolds.

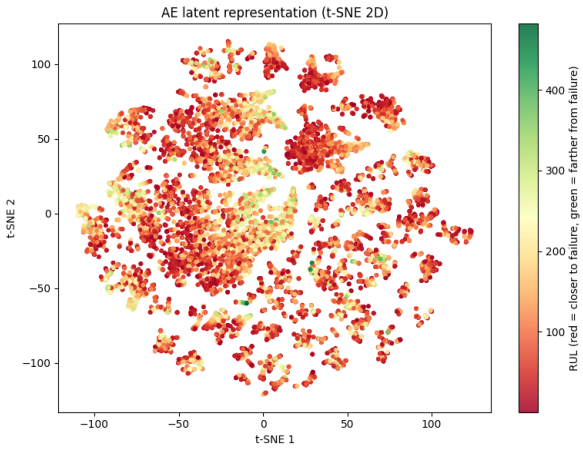

Figure 2. Two-dimensional t-SNE projection of the latent representation learned by the best-performing model (AE) on the Component-X dataset.

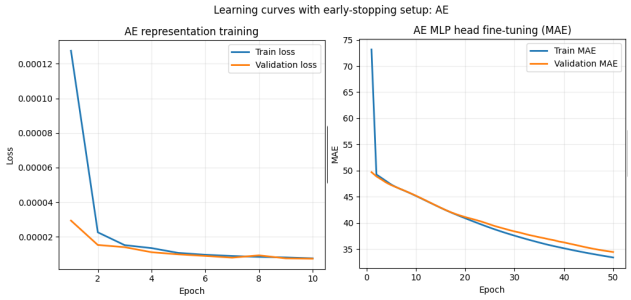

Figure 3. Training and validation curves of the best-performing model for Component-X dataset for both representation learning and MLP head fine-tuning.

Although the representation learned by the VAE appears visually more structured and robust, it achieved the best performance only in the C-MAPSS experiment. We speculate that this may be due to the much lower signal-to-noise ratio in the other two datasets, which could hinder the VAE's ability to learn a stable latent manifold and thereby reduce its overall effectiveness.

In all experiments, the unsupervised learned representations consistently reduced the RUL-head prediction error compared to the baseline models. This demonstrates the effectiveness of leveraging unlabeled telemetry for manifold learning, thereby

Table 3. Final performance comparison on the C-MAPSS dataset in terms of MAE. Values are reported as mean ± standard deviation across five folds.

Method	MAE
Contrastive + MLP	17.6462 ± 0.0707
AE + MLP	18.1267 ± 0.7465
VAE + MLP	$\mathbf{17.2298} \pm 0.2225$
Standalone MLP	21.9369 ± 1.5198
MeanValuePredictor	33.2590 ± 0.0000

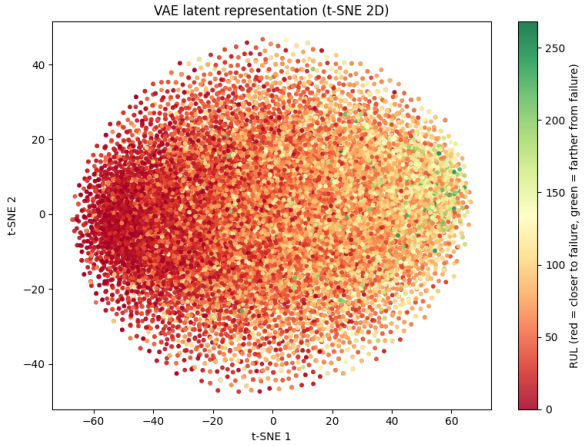

Figure 4. Two-dimensional t-SNE projection of the latent representation learned by the best-performing model (VAE) on the C-MAPSS dataset.

addressing our first research question. It is also worth noting that because the learned representations are substantially smaller than the original feature dimensions, the models that operate on latent vectors require far fewer trainable parameters than the baseline MLP (Table1). This reduction in model size highlights an additional practical advantage of using unsupervised representation learning in RUL prediction.

Across all three experiments, the final prediction errors obtained from the learned representations were relatively similar, indicating no major performance differences among the three methods. Interestingly, each representation learning model i.e. contrastive learning, variational autoencoders, and autoencoders outperformed the others in at least one dataset, meaning no single approach emerged as a clear overall winner. This observation addresses our second research question and suggests that different models excel under different data regimes, reinforcing the idea that the suitability of a representation method depends strongly on the characteristics of the underlying dataset. Furthermore, we note that the performance gap between methods on the Water-Treatment dataset is smaller than on the other two datasets. Our hypothesis is that when training targets are derived from proxy signals rather than true failure events, all methods are equally constrained by the quality of those targets, and the relative advantage of representation learning over the baseline is naturally reduced.

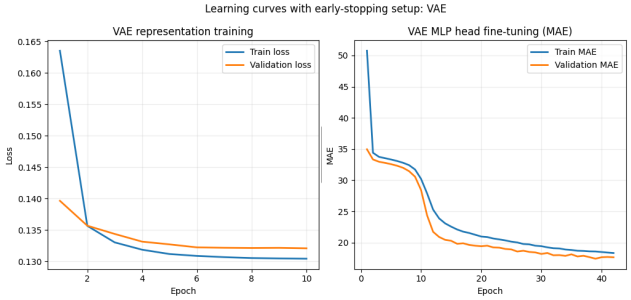

Figure 5. Training and validation curves of the best-performing model for C-MAPSS dataset for both representation learning and MLP head fine-tuning.

Table 4. Final performance comparison on the Water-Treatment pseudo-RUL dataset in terms of MAE. Values are reported as mean ± standard deviation across five folds.

Method	MAE
Contrastive + MLP	$\mathbf{170.5922} \pm 0.5872$
AE + MLP	171.9088 ± 1.3825
VAE + MLP	171.4550 ± 0.7927
Standalone MLP	174.9739 ± 0.6604
MeanValuePredictor	211.6100 ± 0.0000

One notable observation is that contrastive learning consistently exhibits the lowest standard deviation, followed by the variational autoencoder, while the autoencoder shows the highest deviation. This pattern is consistent across all three datasets and suggests that the contrastive representation may be more robust to variations in the data distribution and less sensitive to sampling noise or initialization effects.

Finally, in addressing the third research question, our results indicate that in the absence of maintenance records, pseudo RUL targets derived from soft-failure indicators can serve as a viable alternative. While the effectiveness of this approach ultimately depends on how reliably such soft-failure points are defined, we recommend paying close attention to early degradation indicators, particularly since access to detailed maintenance records is often infeasible in many modern PHM scenarios. Nonetheless, further research is needed to assess the extent to which these signals correlate with actual failure events.

6. CONCLUSION AND FUTURE WORK

Our results show that Pseudo RUL Guided Semi Supervised Learning effectively leverages unlabeled telemetry by combining unsupervised encoders with soft failure based pseudo labels, enabling robust RUL estimation even when failures are sparse or unobserved. Across all datasets, the learned representations formed coherent latent manifolds and consis-

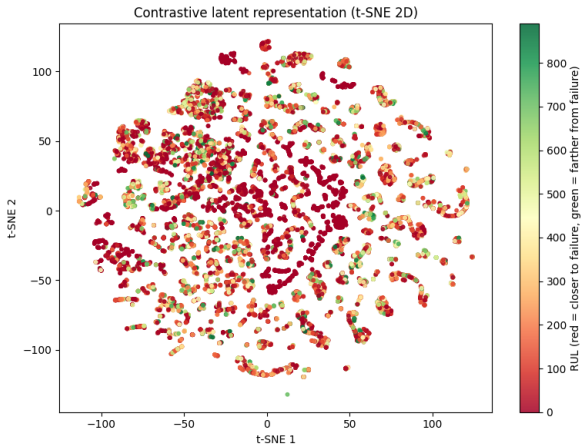

Figure 6. Two-dimensional t-SNE projection of the latent representation learned by the best-performing model (Contrastive) on the Water-Treatment dataset.

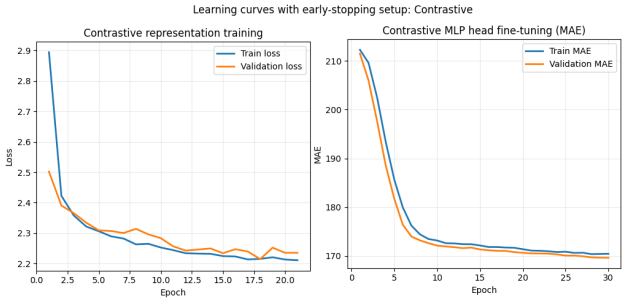

Figure 7. Training and validation curves of the best-performing model for Water-Treatment dataset for both representation learning and MLP head fine-tuning.

tently reduced prediction error relative to raw-feature baselines, demonstrating the value of unlabeled data for RUL prediction under limited supervision. The three representation learning methods each outperformed the others in at least one experiment, indicating that their effectiveness depends on dataset characteristics rather than any inherent superiority. This underscores the need to select representation learning strategies that align with the statistical properties of the underlying telemetry.

Although the learned representations in this study were relatively small, the performance gains achieved by incorporating unsupervised samples suggest that the approach can scale effectively. Building on this idea, we propose training large-scale machine learning models on massive IoT telemetry datasets composed of raw sensory measurements, enabling them to generalize across a broad range of predictive maintenance tasks. Rather than being optimized for a single application—as done here for RUL estimation—such models could learn general-purpose representations that can be efficiently adapted to new tasks with minimal additional training. We hope that the findings presented in this work help move the field in this direction.

ACKNOWLEDGMENT

This research was supported by KK Synergy.

REFERENCES

Alabdallah, A., et al. (2023, September). Discovering Premature Replacements in Predictive Maintenance Time-to-Event Data. *PHM Society Asia-Pacific Conference*, *4*(1). doi: 10.36001/phmap.2023.v4i1.3609

Altarabichi, M. G., et al. (2020). Stacking ensembles of heterogenous classifiers for fault detection in evolving environments. In *30th european safety and reliability conference, esrel 2020 and 15th probabilistic safety assessment and management conference, psam15* (pp. 1068–1068).

Costa, N., et al. (2021). Remaining useful life estimation using a recurrent variational autoencoder. In *International conference on hybrid artificial intelligence systems* (pp. 53–64).

Costa, N., et al. (2022). Variational encoding approach for interpretable assessment of remaining useful life estimation. *Reliability Engineering & System Safety*, *222*, 108353.

Eldele, E., et al. (2021). Time-series representation learning via temporal and contextual contrasting. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*.

He, R., et al. (2022). A semi-supervised gan method for rul prediction using failure and suspension histories. *Mechanical Systems and Signal Processing*, *168*, 108657.

Kamat, P. V., et al. (2021). Deep learning-based anomaly-onset aware remaining useful life estimation of bearings. *PeerJ Computer Science*, *7*, e795.

Kharazian, Z., et al. (2025, March). SCANIA Component X dataset: a real-world multivariate time series dataset for predictive maintenance. *Scientific Data*, *12*(1), 493. doi: 10.1038/s41597-025-04802-6

Kingma, D. P., et al. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krokotsch, T., et al. (2021). Improving semi-supervised learning for remaining useful lifetime estimation through self-supervision. *arXiv preprint arXiv:2108.08721*.

Li, Z., et al. (2024). A survey of deep learning-driven architecture for predictive maintenance. *Engineering applications of artificial intelligence*, *133*, 108285.

Malhotra, P., et al. (2016). Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*.

Rahat, M., et al. (2022, June). Domain Adapta-

tion in Predicting Modeling Turbocharger Failures Using Vehicle's Sensor Measurements. *PHM Society European Conference*, *7*(1), 432–439. doi: 10.36001/phme.2022.v7i1.3340

Rahat, M., et al. (2023, September). Bridging the Gap: A Comparative Analysis of Regressive Remaining Useful Life Prediction and Survival Analysis Methods for Predictive Maintenance. *PHM Society Asia-Pacific Conference*, *4*(1). doi: 10.36001/phmap.2023.v4i1.3646

Rahat, M., et al. (2024, June). SurvLoss: A New Survival Loss Function for Neural Networks to Process Censored Data. *PHM Society European Conference*, *8*(1), 7. doi: 10.36001/phme.2024.v8i1.4052

Revanur, V., et al. (2020). Embeddings Based Parallel Stacked Autoencoder Approach for Dimensionality Reduction and Predictive Maintenance of Vehicles. In *IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning* (Vol. 1325, pp. 127–141). Cham: Springer International Publishing. (Series Title: Communications in Computer and Information Science) doi: 10.1007/978-3-030-66770-$2_1$0

Saxena, A., et al. (n.d.). Damage Propagation Modeling for Aircraft Engine Prognostics.

Star, M., et al. (2025). Dynamical variational autoencoders for estimating the remaining useful life of machinery. *International Journal of Prognostics and Health Management*, *16*(2).

Tonekaboni, S., et al. (2021). Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*.

Wei, Y., et al. (2021). Learning the health index of complex systems using dynamic conditional variational autoencoders. *Reliability Engineering & System Safety*, *216*, 108004.

Yue, Z., et al. (2022). Ts2vec: Towards universal representation of time series. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 36, pp. 8980–8987).

Zerveas, G., et al. (2021). A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th acm sigkdd conference on knowledge discovery & data mining* (pp. 2114–2124).