Fault Prognosis of Turbofan Engines: Eventual Failure Prediction and Remaining Useful Life Estimation

Joseph Cohen[1], Xun Huan[2], and Jun Ni[3]

[1] *Michigan Institute for Data Science, University of Michigan, Ann Arbor, MI, 48109, United States of America*
cohenyo@umich.edu

[2,3] *Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, 48109, United States of America*
xhuan@umich.edu
junni@umich.edu

Abstract

In the era of industrial big data, prognostics and health management is essential to improve the prediction of future failures to minimize inventory, maintenance, and human costs. Used for the 2021 PHM Data Challenge, the new Commercial Modular Aero-Propulsion System Simulation dataset from NASA is an open-source benchmark containing simulated turbofan engine units flown under realistic flight conditions. Deep learning approaches implemented previously for this application attempt to predict the remaining useful life of the engine units, but have not utilized labeled failure mode information, impeding practical usage and explainability. To address these limitations, a new prognostics approach is formulated with a customized loss function to simultaneously predict the current health state, the eventual failing component(s), and the remaining useful life. The proposed method incorporates principal component analysis to orthogonalize statistical time-domain features, which are inputs into supervised regressors such as random forests, extreme random forests, XGBoost, and artificial neural networks. The highest performing algorithm, ANN–Flux with PCA augmentation, achieves AUROC and AUPR scores exceeding 0.94 for each classification on average. In addition to predicting eventual failures with high accuracy, ANN–Flux achieves comparable remaining useful life RMSE for the same test split of the dataset when benchmarked against past work, with significantly less computational cost.

1. Introduction

The field of prognostics and health management (PHM) has attracted recent research attention for large-scale, high-dimensional, and dynamic engineering systems. Typically performed on the component level, the goal of intelligent prognostic approaches is to predict in advance the progression of degradation to facilitate swift and responsible decision-making before catastrophic failure (Lee et al., 2014; Tsui et al., 2015). Typical PHM applications include data-driven fault diagnosis and prognosis of bearing failures (Shao et al., 2018) and gearbox failures (C. Li et al., 2016) utilizing vibration, current, and/or acoustic emission signals. As described by Liao and Köttig (2014), PHM approaches can be separated into different categories: physics-based, expert knowledge-based, or data-driven, with significant potential for hybridization. PHM is essential for reliable operation of safety-critical systems such as nuclear power plants, which have devastating consequences should catastrophic failures occur and are often difficult to predict due to the lack of historical, labeled failure data (Coble et al., 2015).

Recently, deep learning approaches have seen growing popularity in prognostics research, particularly for estimating the remaining useful life (RUL) of physical assets. Graph neural networks (GNNs) and graph convolutional networks (GCNs) have become attractive for fault diagnosis and prognosis tasks with their ability to handle highly correlated and non-Euclidean applications in PHM (T. Li et al., 2022; Lai et al., 2023; Kong et al., 2022). Advanced techniques leveraging long short-term memory (LSTM) networks and attention mechanisms improved RUL prediction for time series data (Song et al., 2022). Berghout et al. (2022) further illustrated how these architectures can benefit from transfer learning to improve prognostic performance.

This paper will focus on the new Commercial Modular Aero-Propulsion System Simulation (N-CMAPSS) dataset, which was featured in the 2021 PHM Data Challenge and centered on accurately estimating the RUL for a small fleet of turbofan engines (Chao et al., 2021b). Openly available from the NASA Prognostics Center of Excellence (PCoE) Data Set

https://doi.org/10.36001/IJPHM.2023.v14i2.3486

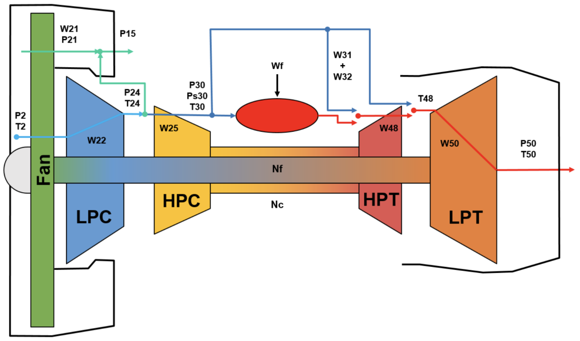

Figure 1. Turbofan engine schematic, courtesy of NASA Prognostics Center of Excellence (Chao et al., 2021b)

Repository (Chao et al., 2021a), N-CMAPSS consists of synthetic run-to-failure trajectories operating under more realistic flight conditions compared to the legacy C-MAPSS dataset used as a popular PHM benchmark (Chao et al., 2021a). The turbofan engines experience multiple failure modes that involve the simultaneous efficiency and/or flow failures of up to 5 rotating subcomponents: fan, low-pressure compressor (LPC), high-pressure compressor (HPC), low-pressure turbine (LPT), and high-pressure turbine (HPT). A schematic representation of a turbofan engine unit is shown in Figure 1.

For predicting the RUL of the turbofan engine units, Lövberg (2021) proposed a neural network-based normalization procedure to effectively denoise the sensor measurements with respect to the dynamic flight conditions. After normalization, the input trajectories were passed into a deep convolutional neural network (CNN) with dilated convolutions in an approach that allows for variable input sequence lengths. Lövberg relied on the provided health state label to sample degraded sequences in their RUL prediction model. DeVol et al. (2021) proposed the integration of inception modules within a CNN architecture to handle the variable trajectory lengths. DeVol et al. reported RUL prediction results using NASA's training-testing split in the N-CMAPSS dataset, streamlining reproducibility and benchmarking for this challenge problem. Solís-Martín et al. (2021) approached this problem by stacking two CNNs in sequence: an encoder model first used for dimensionality reduction and feature extraction, and a secondary model used for RUL prediction. Solís-Martín et al. used Bayesian hyperparameter optimization to tune their models and noted that their prediction results could be improved by reducing overfitting. Biggio et al. (2021) utilized deep Gaussian processes (DGPs) to obtain accurate RUL predictions paired with uncertainty estimates on a subset of N-CMAPSS. Additional studies by Chao et al. (2022) and Berghout et al. (2022) highlighted the potential for physics-based hybridization and transfer learning to improve RUL prediction accuracy.

These approaches benefit from the strengths of deep learning; namely, they allow for effective feature representations to be learned automatically by CNN rather than manually crafted. Clever variations of CNNs, such as Lövberg's approach implementing dilated convolutions and DeVol et al.'s usage of inception modules, have allowed for accurate RUL estimation given varying flight trajectories and input lengths (Lövberg, 2021; DeVol et al., 2021). However, there are key limitations to these approaches that inhibit their potential for practical usage. By focusing solely on RUL prediction, prior methods do not provide a holistic prognosis that predicts the eventual failing component(s). DeVol et al. (2021) mentioned that the resulting RUL predictions lack explainability, and that future work should utilize the labeled failure modes and components provided in the N-CMAPSS dataset to provide a more complete prognosis for turbofan engines.

Our work significantly expands upon past efforts by broadening the research scope to simultaneously predict RUL as well as the eventual failing component(s). Being able to accurately predict and isolate the reason for failure has important implications on maintenance decision-making, equipping operators with the capability to dispatch the appropriate experts and resources in a timely manner. Such predictive maintenance strategies can enable intelligent inventory optimization (Bousdekis et al., 2017) and reduce reactive maintenance costs, which may account for up to 40% of the overall budget in large industries (Bagavathiappan et al., 2013). Previous work attempting to perform fault classification as well as RUL prediction typically carry these objectives in separate stages, with Gupta et al. (2023) devising a method to perform fault classification after RUL prediction, whereas J. Y. Wu et al. (2021) first classified the health state and then performed RUL regression. Additionally, X. Wu and Ye (2016) investigated RUL estimation for solid oxide fuel cells following a separate fault detection stage. To the best of our knowledge, no previous studies have attempted to explicitly optimize RUL regression and component-level classification on a simultaneous basis on a dataset as large and high-dimensional as N-CMAPSS. This is the first attempt at a unified model to effectively accomplish fault detection, isolation, and RUL estimation for the N-CMAPSS benchmark dataset.

To maximize applicability for a real-world scenario, we aim to simultaneously predict three meaningful indicators: 1) the current health state; 2) the eventual failing component(s); and 3) the RUL until catastrophic failure. We accomplish these goals by first simplifying the feature extraction process to enable comparisons amongst state-of-the-art machine learning regressors. Then, we derive and optimize a specialized loss function that balances classification and regression objectives. We also compare the performance of state-of-the-art machine learning regressors and important pre-processing steps such as orthogonalization via principal components analysis (PCA). Our main contributions for this research effort are summarized as follows:

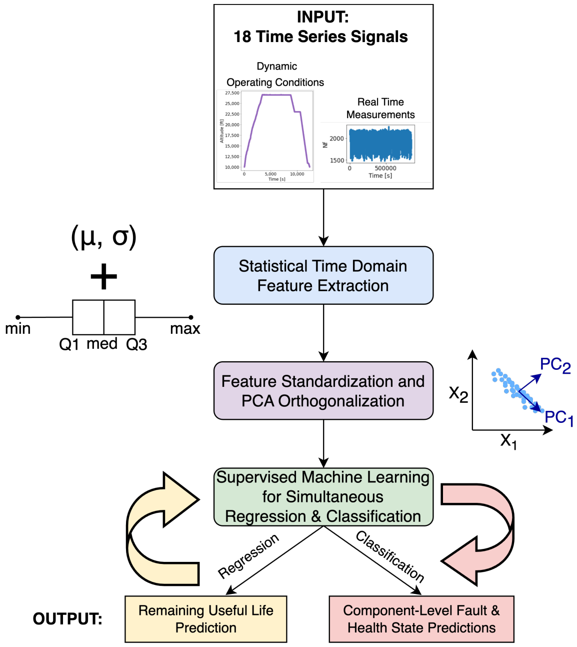

Figure 2. Proposed methodology flow for failure prediction

1. Reformulating and expanding upon the 2021 PHM Data Challenge to include health state detection and forecasting eventual failures;
2. Deriving a customized loss function to simultaneously optimize classification and regression PHM objectives; and
3. Accurately predicting health state, eventual failures, and RUL with state-of-the-art regression approaches benchmarked with prior work.

In the following sections of the paper, we will detail our proposed methodology, our reproducible results, and provide comparisons to previous work.

2. METHODS

First, we will describe the N-CMAPSS dataset, and introduce the input variables and dataset composition in detail. With the expanded goal to predict the current health state and eventual failing components in addition to RUL, our proposed methodology encompasses both classification and regression objectives. Our method is summarized in three steps: 1) feature extraction; 2) feature normalization and orthogonalization via PCA; and 3) training a supervised machine learning model to obtain the final predictions. Figure 2 provides an overview of the flow for the proposed methodology.

2.1. Dataset Description

The N-CMAPSS dataset consists of 8 provided subsets and contains 90 engine units in total. In our research, we combine flow and efficiency failures into one general failure category for each mechanical component. Table 1 provides a summary of the failure modes present in each subset. In the dataset, engine units have a lifetime rated typically between 60 and 100 cycles, with the overall objective being to estimate the RUL until catastrophic failure. Each flight cycle is of variable length and is characterized by 18 time series signals: 4 flight data descriptors $W = \{W_1, W_2, W_3, W_4\}$ summarizing the dynamic operating conditions, and 14 real-time sensor measurements $X_s = \{X_{s_1}, X_{s_2}, \ldots, X_{s_{14}}\}$. In addition to the time series signals, each cycle also includes auxiliary variables $A = \{A_1, A_2, A_3, A_4\}$ useful for understanding the context of a flight cycle: the unit number, cycle number, a categorical flight class variable F_c representing the length of the flight (set to 1 for short flights, 2 for medium flights, and 3 for long flights), as well as a binary health state variable h_s (set to 1 for healthy status and 0 for unhealthy status). We note that the simulated engines are flown past unhealthy operation until end of life (i.e., catastrophic failure). Table 2 provides a summary of the variables provided in the dataset. In all, there are a total of 6825 flight cycles in the dataset, with engines averaging approximately 75 cycles per unit.

Table 1. Component-level failure descriptions as labeled from the N-CMAPSS dataset (Chao et al., 2021b)

Subset Name	Units	Fan Fail?	LPC Fail?	HPC Fail?	HPT Fail?	LPT Fail?
DS01	10	No	No	No	Yes	No
DS03	15	No	No	No	Yes	Yes
DS04	10	Yes	No	No	No	No
DS05	10	No	No	Yes	No	No
DS06	10	No	Yes	Yes	No	No
DS07	10	No	No	No	No	Yes
DS08a	15	Yes	Yes	Yes	Yes	Yes
DS08c	10	Yes	Yes	Yes	Yes	Yes

2.2. Feature Extraction

Feature extraction is necessary to reduce the input dimensionality of the dataset. Although there are only 90 turbofan engine units in the N-CMAPSS dataset as per Table 1, the dataset contains over 63 million timestamps and requires reduction for subsequent data processing. As in previous work, we aim to make predictions on a per-cycle basis (DeVol et al., 2021).

In this study, we extract cycle-wide statistical time domain features to summarize the distribution for each time series. These features include:

1. Mean;
2. Standard deviation;
3. Minimum;
4. 1st Quartile;
5. Median;
6. 3rd Quartile;
7. Maximum,

Table 2. Auxiliary, flight descriptors, and sensor measurement variables used in N-CMAPSS dataset (Chao et al., 2021b)

Variable	Symbol	Description	Units
A_1	unit	Unit number	-
A_2	cycle	Flight cycle number	-
A_3	F_c	Flight class	-
A_4	h_s	Health state	-
W_1	alt	Altitude	ft
W_2	Mach	Mach number	-
W_3	TRA	Throttle-Resolver angle	%
W_4	T2	Total temp. at fan inlet	°R
X_{s_1}	Wf	Fuel flow	pps
X_{s_2}	Nf	Physical fan speed	rpm
X_{s_3}	Nc	Physical core speed	rpm
X_{s_4}	T24	Total temp. at LPC outlet	°R
X_{s_5}	T30	Total temp. at HPC outlet	°R
X_{s_6}	T48	Total temp. at HPT outlet	°R
X_{s_7}	T50	Total temp. at LPT outlet	°R
X_{s_8}	P15	Total pressure in bypass-duct	psia
X_{s_9}	P2	Total pressure at fan inlet	psia
$X_{s_{10}}$	P21	Total pressure at fan outlet	psia
$X_{s_{11}}$	P24	Total pressure at LPC outlet	psia
$X_{s_{12}}$	Ps30	Static pressure at HPC outlet	psia
$X_{s_{13}}$	P40	Total pressure at burner outlet	psia
$X_{s_{14}}$	P50	Total pressure at LPT outlet	psia

for all 18 time series signals (the 4 W's and 14 X_s's), resulting in 126 statistical features. We append this feature set by additionally selecting features that are held constant per cycle such as:

1. Time duration of cycle;
2. Current cycle number (A_2);
3. Flight class (A_3).

Importantly, we do not use the unit number A_1 and health state A_4 as input features, and we opt to learn the health state as an output instead. All together, this results in 129 total features extracted from the N-CMAPSS dataset. In more general terms, this feature extraction method is applied for n training cycles, with $\mathbf{x}_j \in \mathbb{R}^n$ representing the vector of samples for the j^{th} feature. Finally, the feature vectors are concatenated into a single data matrix containing all p features, $\mathbf{X} \in \mathbb{R}^{n \times p}$.

2.3. Feature Normalization and PCA Orthogonalization

Features extracted from the time series signals may be of different scales and units. As a result, normalization helps ensure that predictions are not influenced by these differences. First, we apply a min-max normalization scheme across all features to map all features in the bounded range $[0, 1]$ as shown in Eq. (1):

$$\bar{\mathbf{x}}_j = \frac{\mathbf{x}_j - \min(\mathbf{x}_j)}{\max(\mathbf{x}_j) - \min(\mathbf{x}_j)}. \tag{1}$$

Once again, we concatenate the feature vectors into a normalized data matrix, $\bar{\mathbf{X}} \in \mathbb{R}^{n \times p}$. After obtaining the normalized data matrix, PCA orthogonalization is recommended as a multivariate preprocessing step to obtain a set of uncorrelated variables. PCA is typically used to achieve dimension reduction by retaining the most important principal components (PCs) such that the explained variance is maximized (Jollife & Cadima, 2016). However, we have found that in practice, there is utility to keeping all PCs to improve training results. This is potentially because the features extracted are significantly correlated, and therefore simply using PCA for its orthogonalization benefits may improve the performance of gradient descent-based optimization methods employed in training. In Section 3, we will compare results with and without PCA orthogonalization for all models. PCA can be formulated as a linear transformation using the eigendecomposition of the sample correlation matrix $\mathbf{Q} \in \mathbb{R}^{p \times p}$ of the features from $\bar{\mathbf{X}}$, as shown in Eqs. (2)–(3):

$$\begin{aligned} \mathbf{Q} &= \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \\ &= \begin{bmatrix} v_1 & \dots & v_p \end{bmatrix} \begin{bmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_p \end{bmatrix} \begin{bmatrix} v_1 & \dots & v_p \end{bmatrix}^T \qquad (2) \\ \tilde{\mathbf{X}} &= \bar{\mathbf{X}}\mathbf{V} = \begin{bmatrix} \bar{\mathbf{x}}_1 & \dots & \bar{\mathbf{x}}_p \end{bmatrix} \begin{bmatrix} v_1 & \dots & v_p \end{bmatrix}, \qquad (3) \end{aligned}$$

where $v_1, v_2, \dots, v_p$ are the PCs with corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$. The resulting matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times p}$ is the newly orthogonalized training dataset scored along the PC axes.

2.4. Output Labeling Scheme

While N-CMAPSS provides h_s and failure mode information as possible inputs for a RUL prediction model, we aim to instead predict them as outputs encoded as binary variables. As mentioned previously, these additional outputs will provide a more comprehensive prognosis of the degraded turbofan engine unit. With these new outputs, we require a labeling scheme for training a model. For learning the current cycle health state h_s, we borrow the labels provided in the N-CMAPSS dataset, i.e., a label of "1" for healthy operation and "0" for unhealthy operation. We introduce a vector of possible eventual failures $\mathbf{y}_{EF} = \begin{bmatrix} y_{Fan} & y_{LPC} & y_{HPC} & y_{HPT} & y_{LPT} \end{bmatrix}^T$, in which each variable $\mathbf{y}_{comp} \in \mathbf{y}_{EF}$ is binary, with the positive label indicating eventual failure as specified in Table 1. For example, for the DS06 subset in which the LPC and HPC components eventually fail (even if the engine is presently healthy), $\mathbf{y}_{EF} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix}^T$. Lastly, for the RUL training label, we follow the N-CMAPSS convention, which provides RUL $\in \mathbb{Z}^*$ as calculated by subtracting the current cycle number from the total lifetime of the engine unit, i.e., RUL $= t_{EOL} - A_2$. With these definitions, we can prepare the ground truth vector of labels $\mathbf{y} = \begin{bmatrix} h_s & \mathbf{y}_{EF}^T & \text{RUL} \end{bmatrix}^T$ paired with the features of each cycle.

2.5. Training Loss Function and Model Evaluation

Handling classification and regression objectives simultaneously provides additional complexity for training a predictive machine learning model. **We propose optimizing a customized loss function that explicitly combines both objectives**. First, we base the RUL loss contribution from NASA's scoring criteria (Chao et al., 2021b), which penalizes overestimation of RUL to favor conservative predictions and is defined in Eqs. (4)–(6):

$$s_c\left(\mathrm{RUL}, \widehat{\mathrm{RUL}}\right) = \frac{1}{n}\sum_{i=1}^{n} \exp\left(\alpha|\mathrm{RUL}_i - \widehat{\mathrm{RUL}}_i|\right) - 1 \quad (4)$$

$$\mathrm{RMSE}\left(\mathrm{RUL}, \widehat{\mathrm{RUL}}\right) = \left(\frac{1}{n}\sum_{i=1}^{n}\left(\mathrm{RUL}_i - \widehat{\mathrm{RUL}}_i\right)^2\right)^{1/2} \quad (5)$$

$$\mathrm{NASA}\left(\mathrm{RUL}, \widehat{\mathrm{RUL}}\right) = 0.5\,\mathrm{RMSE} + 0.5 s_c, \quad (6)$$

in which α is the overestimation penalty equal to 1/13 if the RUL is underestimated (i.e., $\widehat{\mathrm{RUL}}_i < \mathrm{RUL}_i$) and equal to 1/10 for overestimations. We note that this makes the loss function non-differentiable. However, modern automatic differentiation packages are able to estimate gradients via subgradients and are more flexible for handling unconventional loss functions (Innes et al., 2019). In our work, we substitute the values of α directly and rewrite Eq. (4) as follows using an indicator function, which allows for easier implementation within automatic differentiation coding environments:

$$u = \left(\frac{1}{13} + \frac{3}{130}\mathbf{1}_{\widehat{\mathrm{RUL}}_i > \mathrm{RUL}_i}\right)|\mathrm{RUL}_i - \widehat{\mathrm{RUL}}_i| \quad (7)$$

$$s_c\left(\mathrm{RUL}, \widehat{\mathrm{RUL}}\right) = \frac{1}{n}\sum_{i=1}^{n}\exp(u) - 1. \quad (8)$$

This alternative formulation essentially "upgrades" the α penalty from a base value of 1/13 to 1/10 when RUL is overestimated.

In addition to the loss in Eq. (6) for capturing RUL errors, we further incorporate the binary cross-entropy loss $\mathrm{BCE}\left(\mathbf{y}_{\setminus \mathrm{RUL}}, \hat{\mathbf{y}}_{\setminus \widehat{\mathrm{RUL}}}\right)$ to reflect the errors on the q classification outputs. Together, we obtain an overall loss function through a weighted sum of the terms introduced above:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = \mathrm{NASA}\left(\mathrm{RUL}, \widehat{\mathrm{RUL}}\right) + \gamma \mathrm{BCE}\left(\mathbf{y}_{\setminus \mathrm{RUL}}, \hat{\mathbf{y}}_{\setminus \widehat{\mathrm{RUL}}}\right), \quad (9)$$

where γ is a tunable scalar weight coefficient. Since the NASA score is based on RUL regression, this term tends to carry a larger magnitude compared to the BCE. In order to achieve a more balanced contribution of the regression and classification parts, we set $\gamma = 10$ for our test cases, although other values can be selected to reflect the relative importance of the regression and classification tasks under the particular PHM situation being considered.

Additionally, because benchmarking comparisons between multiple machine learning regressors have not yet been provided in previous work on the N-CMAPSS dataset (DeVol et al., 2021; Lövberg, 2021; Solís-Martín et al., 2021), we compare the performance of four state-of-the-art machine learning methods: random forests (RFs) (Genuer et al., 2017), extreme random forests (ERFs) (also known as extra trees) (Maier et al., 2015), XGBoost (XGB) (Chen & Guestrin, 2016), and artificial neural networks (ANNs) (Mahamad et al., 2010). Specifically, we will compare the performance of the tree-based ensemble regressors trained to minimize mean squared error (MSE) to an ANN that minimizes our proposed loss function in Eq. (9). For completeness, we will also compare these results to an ANN minimizing MSE. To evaluate the quality of classification predictions, we will use the area under receiver operating characteristics (AUROC) and precision-recall curves (AUPR) metrics to evaluate performance at all possible thresholds. Meanwhile, root-mean-square error (RMSE), the NASA scoring function detailed in Eq. (6), mean absolute error (MAE), and MAE normalized as a percentage of the unit's lifetime will be reported for judging regression quality for RUL predictions.

3. EXPERIMENTAL SETUP

For reproducibility, results are reported using the built-in N-CMAPSS dataset split, as in past work by DeVol et al. (2021). This dataset split is notable for having a testing set with entire engine units that are unseen in the training set, making the benchmark problem more realistic and challenging. The split follows an approximately 60%-40% training-testing ratio, with 4089 cycles in the training set and 2736 cycles in the test set. The training set is further divided with a 80%-20% training-validation split, with a randomly selected holdout validation set used for early stopping and hyperparameter tuning. Following the feature extraction method detailed in Section 2.2, we employ 129 features. Using the min-max normalization and PCA orthogonalization methods from the popular scikit-learn package (Pedregosa et al., 2012), we normalize the training set and apply these learned transformations to the testing set.

To prepare the regressors minimizing the MSE loss function, it is necessary to scale the labels such that the RUL regression error does not dominate the MSE calculation. This is done by simply multiplying the binary encoded labels in $\mathbf{y}$ by 100, thereby putting the binary labels in the same magnitudes as the RUL labels. After training the models, the AUROC and AUPR metrics are then computed using the resulting classification predictions on the test set in $\hat{\mathbf{y}}$ to serve as robust indicators of performance averaged across all possible thresholds. Table 3 shows the classification and regression results for RF, ERF, XGB, an ANN also trained on MSE (ANN–MSE), and an ANN trained on the custom loss function derived in Section 2.5 (ANN–Flux). The average and standard deviation for the performance on the test set are reported after repeating with 4 randomized validation sets. Results with and without the PCA orthogonalization step are also included, demonstrating the impact of the preprocessing procedure on minimally tuned models. The RF and ERF regressors, implemented using scikit-learn, each contain 100 base estimators. The XGBoost method also uses 100 estimators, with the learning rate η set to 0.3 and the max depth of a tree set to 6.

Both ANNs share the same shallow architecture of two hidden layers with 64 and 32 neurons each with ReLU activations, employing the ADAM optimizer and trained for 5000 epochs with a batch size of 256. All methods are implemented using the Julia 1.8.5 programming language (Bezanson et al., 2014) and both ANNs are designed using the Flux deep learning backend, which allows for auto-differentiation of custom loss functions (Innes, 2018; Innes et al., 2019). The results in Table 3 may be further improved with a more thorough hyperparameter optimization and merely illustrate the potential for the simultaneous prediction of eventual failures alongside RUL.

Benchmarked on a local MacBook Pro machine running macOS Ventura 13.2.1 with Apple M2 Max CPU and 32 GB of RAM, it takes approximately 60 seconds total to train and evaluate the RF, ERF, and XGB methods. On the same processor, the Flux models take approximately 250 seconds to train. The feature extraction is the longest step in terms of runtime, taking around 300 seconds to load the dataset and extract all 129 features for the training and testing sets.

4. RESULTS

The ANN–Flux method with the PCA preprocessing step accurately predicts the current health state and the eventual failure component(s) with AUROC and AUPR scores exceeding 0.95 for each output. This is especially notable considering the significant overlap of the failing components depending on the failure modes (see Table 1). In addition, the RUL prediction also outperforms the other techniques tested. The parity plot in Figure 3**a**) visualizes the ANN–Flux RUL predictions in the testing set versus the ground truth labels. In addition, producing a figure similar to DeVol et al. (2021), Figure 3**b**) also illustrates the ANN–Flux RUL predictions with the ground truth RUL labels sorted from least-to-greatest.

We note that these RUL predictions are directly output from the ANN–Flux model and further considerations may improve their quality and usefulness in practice. For example, despite the asymmetric NASA scoring function favoring conservative underestimates, the average prediction error still slightly overestimates the ground truth by 0.65 cycles. This contrasts with the training prediction error, which on average underestimates the ground truth by 0.50 cycles. Further adjustments on the overestimation penalty α and the classification loss weight γ may skew the prediction error towards underestimation. In addition, we have not instituted hard constraints to guarantee nonnegative RUL values. Post-processing transformations such as the ReLU function can be implemented in the future to rectify the outputs such that all resulting RUL predictions are nonnegative.

Notably, the PCA orthogonalization pre-processing step has a profound impact on classification performance for the eventual failure of the mechanical components. These findings are consistent among all attempted machine learning methods. However, PCA orthogonalization did not appear to improve the regression performance in the same way; 3 out of the 5 attempted methods had increased RMSE when inputs were orthogonalized. This suggests that using PCA to orthogonalize these extracted features is especially useful for binary classification predictions, but may not always lead to better results for minimizing RUL error.

Having additional classification outputs enables explainable analysis of RUL predictions along various slices of the dataset. For example, Figure 4 illustrates the RUL prediction errors for unhealthy versus healthy cycles. Intuitively, the interquartile range for unhealthy operating cycles is substantially narrower, indicating that RUL predictions on average improve throughout the life of the engine unit.

It is also useful to determine whether there are certain components with higher variance in RUL prediction errors; by observing the RUL prediction errors on a per-component basis, operators can glean more information and make targeted decisions based on their confidence of the prognosis. Similar to Figure 4, Figure 5 plots the RUL prediction error spread of the test set for each of the labeled eventual mechanical component failures. Figure 5 demonstrates that the RUL prediction errors have a median centered near 0 for each mechanical component and there is no significant component-based bias identified. Relatively, the compressor failures have a tighter concentration around 0 and the turbine failures are more negatively skewed, indicating more underestimates, but we note that it is difficult to draw definitive conclusions due to overlapping failures.

5. DISCUSSION

Our findings have broad economic implications beyond engine prognostics, as a similar approach could potentially be applied for other PHM applications. Our approach is enabled by expanding the formulation of the 2021 PHM Data Challenge to simultaneously include classification and regression objectives, taking full advantage of the provided labels in the N-CMAPSS dataset. However, we note that

Table 3. Classification and regression results for N-CMAPSS dataset for ensemble methods, with the PCA-orthogonalized XGBoost method generally outperforming the other benchmark methods

Output	Metric	RF	+ PCA	ERF	+ PCA	XGB	+ PCA
Health	AUROC	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00
State	AUPR	0.99 ± 0.00	0.98 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.98 ± 0.01
Fan	AUROC	0.79 ± 0.01	0.91 ± 0.01	0.78 ± 0.00	0.91 ± 0.00	0.88 ± 0.00	0.95 ± 0.03
Failure	AUPR	0.63 ± 0.02	0.87 ± 0.01	0.60 ± 0.02	0.86 ± 0.01	0.78 ± 0.02	0.90 ± 0.03
LPC	AUROC	0.79 ± 0.01	0.90 ± 0.01	0.78 ± 0.01	0.89 ± 0.00	0.88 ± 0.00	0.94 ± 0.03
Failure	AUPR	0.63 ± 0.01	0.81 ± 0.01	0.62 ± 0.01	0.78 ± 0.01	0.81 ± 0.01	0.90 ± 0.05
HPC	AUROC	0.83 ± 0.01	0.92 ± 0.00	0.82 ± 0.01	0.92 ± 0.01	0.93 ± 0.01	0.96 ± 0.03
Failure	AUPR	0.79 ± 0.01	0.90 ± 0.01	0.79 ± 0.01	0.89 ± 0.01	0.93 ± 0.01	0.96 ± 0.03
HPT	AUROC	0.80 ± 0.01	0.88 ± 0.00	0.80 ± 0.01	0.88 ± 0.00	0.88 ± 0.00	0.91 ± 0.04
Failure	AUPR	0.77 ± 0.01	0.86 ± 0.00	0.77 ± 0.01	0.85 ± 0.00	0.87 ± 0.01	0.90 ± 0.03
LPT	AUROC	0.78 ± 0.01	0.87 ± 0.00	0.77 ± 0.01	0.87 ± 0.00	0.85 ± 0.01	0.90 ± 0.03
Failure	AUPR	0.76 ± 0.01	0.84 ± 0.00	0.76 ± 0.01	0.84 ± 0.00	0.85 ± 0.01	0.90 ± 0.04
RUL	RMSE	10.30 ± 0.07	10.97 ± 0.02	10.34 ± 0.12	10.34 ± 0.03	9.97 ± 0.05	9.73 ± 0.86
	NASA	5.92 ± 0.05	6.37 ± 0.02	5.95 ± 0.08	5.96 ± 0.02	5.72 ± 0.03	5.58 ± 0.52
	MAE (cycles)	8.13 ± 0.04	8.83 ± 0.03	8.14 ± 0.10	8.26 ± 0.04	7.66 ± 0.04	7.43 ± 0.67
	MAE (%)	11.01 ± 0.06	11.84 ± 0.03	11.06 ± 0.14	11.08 ± 0.05	10.32 ± 0.05	9.97 ± 1.00

Table 4. Classification and regression results for N-CMAPSS dataset for ANN methods, with the proposed PCA-orthogonalized ANN–Flux method achieving balanced classification and regression performance compared to ANN–MSE and ANN–Flux without PCA pre-processing

Output	Metric	ANN–MSE	+ PCA	ANN–Flux	+ PCA
Health	AUROC	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00
State	AUPR	0.99 ± 0.00	0.98 ± 0.01	0.99 ± 0.00	0.99 ± 0.00
Fan	AUROC	0.95 ± 0.00	0.99 ± 0.01	0.95 ± 0.02	0.98 ± 0.00
Failure	AUPR	0.93 ± 0.01	0.98 ± 0.04	0.93 ± 0.03	0.96 ± 0.01
LPC	AUROC	0.93 ± 0.01	0.99 ± 0.00	0.93 ± 0.02	0.97 ± 0.00
Failure	AUPR	0.87 ± 0.02	0.97 ± 0.01	0.86 ± 0.05	0.95 ± 0.01
HPC	AUROC	0.98 ± 0.00	0.99 ± 0.00	0.98 ± 0.00	0.99 ± 0.00
Failure	AUPR	0.98 ± 0.00	0.99 ± 0.00	0.97 ± 0.01	0.99 ± 0.00
HPT	AUROC	0.95 ± 0.01	0.98 ± 0.01	0.95 ± 0.01	0.97 ± 0.01
Failure	AUPR	0.96 ± 0.01	0.98 ± 0.01	0.96 ± 0.01	0.97 ± 0.01
LPT	AUROC	0.89 ± 0.01	0.96 ± 0.00	0.90 ± 0.03	0.94 ± 0.02
Failure	AUPR	0.89 ± 0.01	0.96 ± 0.01	0.91 ± 0.03	0.94 ± 0.02
RUL	RMSE	7.81 ± 0.24	9.01 ± 0.19	8.14 ± 0.30	8.20 ± 0.17
	NASA	4.36 ± 0.15	13.89 ± 8.20	4.57 ± 0.16	4.68 ± 0.15
	MAE (cycles)	5.88 ± 0.21	6.48 ± 0.15	6.09 ± 0.13	6.16 ± 0.09
	MAE (%)	7.67 ± 0.27	8.52 ± 0.22	8.06 ± 0.10	8.19 ± 0.15

our work targets predicting eventual failures broken down to the *detailed component-level* rather than the 7 pre-defined *higher-level failure modes*. Predicting component-level failure is a more challenging generalization to predicting failure modes, since the solution space is expanded to allow all 2^5 possible combinations of failing components, even though the dataset is only sparsely populated by the 7 pre-defined failure modes. The benefit of this generalization is that a successful model would learn how the same components can fail with dramatically different failure data, which has practical implications for maintenance decision-making and inventory costs. Previous research on this dataset also utilized the labeled health state as an input to improve RUL predictions (Lövberg, 2021); we have relaxed assumptions by instead learning the health state as an additional output.

The computation effort of our approach compared to past work is also noteworthy. Table 5 compares our model's size (in terms of number of trainable parameters) and the obtained RUL RMSE in comparison to the state-of-the-art methods from available literature. Here, we note that not all prior work reported the number of trainable parameters, and several works did not report RUL results on the same test split encompassing the entire N-CMAPSS dataset. The joint classification-regression approach taken in this paper also constrains the solution in terms of RUL performance. **While it is not the highest performing method when solely pursuing RUL regression, ANN–Flux aims to tell us "why", and not just "when"**.

ANN–Flux is also remarkably simple, with number of parameters approximately two orders of magnitude fewer compared to the deep CNNs of prior work. In a realistic scenario with larger datasets, smaller networks are less expensive to run in real-time, streamlining inferencing efforts. Although our method requires hand-selected features prior to training an ANN, the extracted features are simple statistical features and do not require significant domain expertise. Perhaps surprisingly, predicting RUL in addition to the eventual failure component(s) and current health state does not significantly

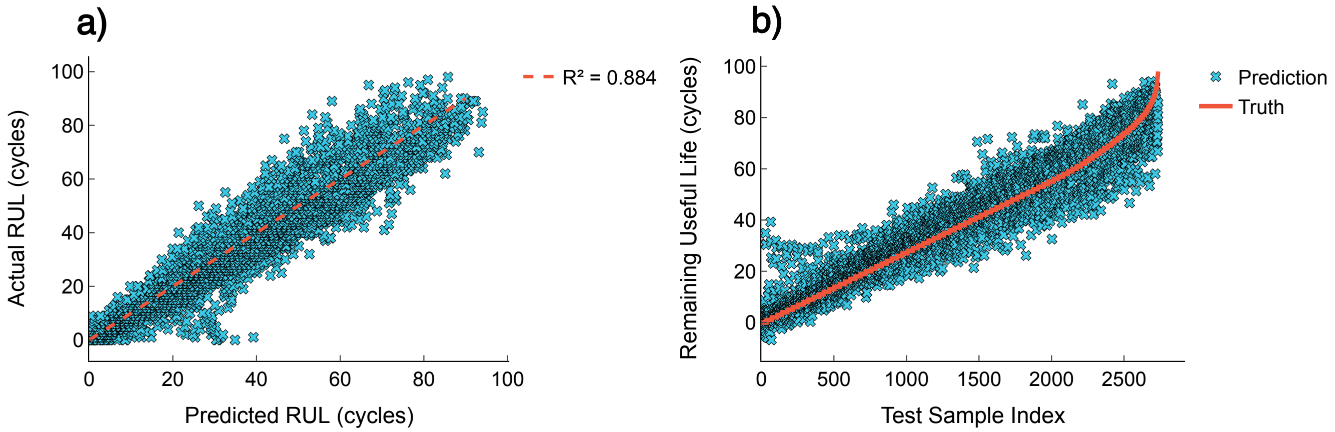

Figure 3. **a)** Parity plot comparing actual and predicted RUL values for ANN–Flux predictions on the N-CMAPSS testing engine unit set; **b)** ANN–Flux predictions scatter with ground truth labels sorted from least-to-greatest

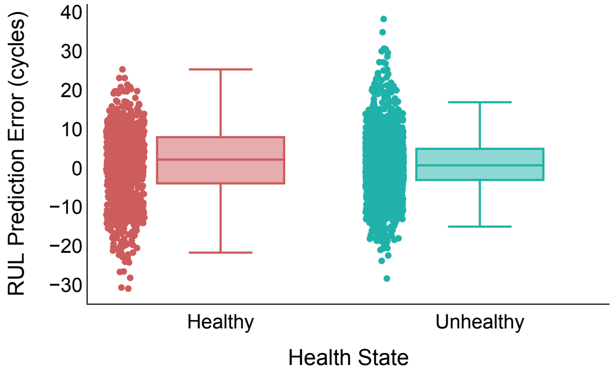

Figure 4. Box-and-whisker plot for RUL error for healthy and unhealthy operation cycles

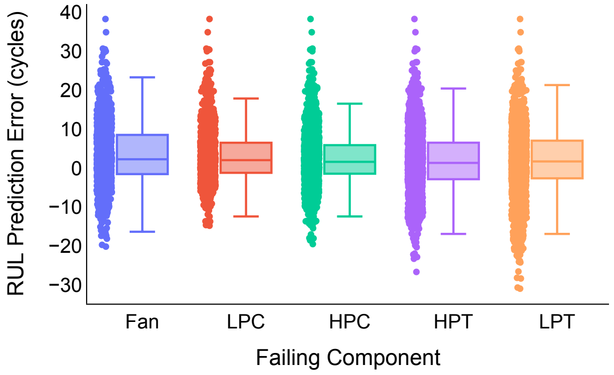

Figure 5. Box-and-whisker plots for RUL prediction errors for each eventual failing component

sacrifice the RUL prediction error.

Table 5. Number of trainable parameters and RUL RMSE compared with state-of-the-art

Approach	# Trainable Parameters	RUL RMSE
(DeVol et al., 2021)	1,030,000	12.5
(Solís-Martín et al., 2021)	4,089,465	6.24
(Biggio et al., 2021)	169,401	7.31
(Song et al., 2022)	N/A	6.867
(Berghout et al., 2022)	N/A	5.64
ANN–Flux + PCA	**10,631**	**8.20 ± 0.17**

To the authors' knowledge, our work is also the first to compare multiple state-of-the-art regression approaches for predicting component failures and RUL estimation for the N-CMAPSS benchmark. We also provide comparisons with and without PCA orthogonalization and for multiple loss functions, with tangible improvements for both RUL and binary classifications with these computational approaches. We hope that our contributions encourage future benchmarking efforts on the N-CMAPSS dataset and for PHM research.

Despite these advancements, important limitations remain that require addressing in future work. Firstly, while RUL prediction is comparable with past work, the prediction errors still have a large variance. Integration with physical modeling is suggested in the future to improve the confidence of RUL predictions. Moreover, failure data are difficult to obtain in practice, and as a result, industrial datasets are often imbalanced (Santos et al., 2018), threatening the utility of fully supervised learning techniques. As a result, more research is required in semi-supervised and unsupervised methods to at least lighten the supervision requirement for AI algorithms to provide accurate prognoses. In addition, while PCA orthogonalization vastly improved the component failure predictions, the derived PC variables lack physical meaning, hindering the explainability of the input features. This step makes the current formulation incompatible with explainable AI (XAI) methods such as SHAP, which attempt to explain black-box model predictions in terms of additive marginal contributions of features (Senoner et al., 2022). While being able to accurately isolate eventual failures on a component-level provides inherent explainability compared to previous efforts, we leave XAI integration for future work.

6. CONCLUSION

Our work as benchmarked on the N-CMAPSS dataset uniquely demonstrates the potential for an approach that simultaneously detects the current health state, predicts which component(s) will fail, and then estimates the number of cycles until failure. In essence, this integrates the important disciplines of anomaly detection and fault diagnosis—conventionally requiring multiple models—in one prognostic model that makes accurate predictions, even for presently healthy units. Our main contributions and findings for this research effort are restated as follows:

1. Reformulated and expanded the scope of the 2021 PHM Data Challenge to include health state detection and eventual failure prognosis;
2. Customized loss function derived to simultaneously combine classification and regression objectives;
3. Accurately predicted health state and eventual failures, with AUROC and AUPR exceeding 0.94 on average for each classification prediction accomplished with the ANN–Flux methodology; and
4. Comparable RUL RMSE achieved for the same dataset split and with less computational effort required for training when benchmarked against prior work.

The authors hope that these contributions will help bolster PHM research and Industry 4.0 efforts to improve safety, lower costs, and enhance decision-making in the age of Big Data.

DATA AVAILABILITY

We plan on making all code for this paper fully available on GitHub for maximum transparency and encourage reproducibility to further N-CMAPSS as a benchmark for PHM research. The N-CMAPSS dataset is publicly available for download in NASA's Prognostics Center of Excellence Data Repository: https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository.

ACKNOWLEDGMENT

The authors would like to thank the S. M. Wu Manufacturing Research Center as well as the Uncertainty Quantification & Scientific Machine Learning Group at the University of Michigan for invaluable feedback and support.

REFERENCES

Bagavathiappan, S., Lahiri, B. B., Saravanan, T., Philip, J., & Jayakumar, T. (2013). Infrared thermography for condition monitoring – A review. *Infrared Physics & Technology*, *60*, 35–55. doi: 10.1016/J.INFRARED.2013.03.006

Berghout, T., Mouss, M. D., Mouss, L. H., & Benbouzid, M. (2022, 12). ProgNet: A Transferable Deep Network for Aircraft Engine Damage Propagation Prognosis under Real Flight Conditions. *Aerospace 2023*, *10*, 10. doi: 10.3390/AEROSPACE10010010

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2014). Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, *59*, 65–98.

Biggio, L., Wieland, A., Chao, M. A., Kastanis, I., & Fink, O. (2021). Uncertainty-Aware Prognosis via Deep Gaussian Process. *IEEE Access*, *9*, 123527. doi: 10.1109/ACCESS.2021.3110049

Bousdekis, A., Papageorgiou, N., Magoutas, B., Apostolou, D., & Mentzas, G. (2017). A Proactive Event-driven Decision Model for Joint Equipment Predictive Maintenance and Spare Parts Inventory Optimization. *Procedia CIRP*, *59*, 184–189. doi: 10.1016/J.PROCIR.2016.09.015

Chao, M. A., Kulkarni, C., Goebel, K., & Fink, O. (2021a). Aircraft Engine Run-To-Failure Dataset Under Real Flight Conditions. *NASA Ames Prognostics Data Repository*.

Chao, M. A., Kulkarni, C., Goebel, K., & Fink, O. (2021b). PHM Society Data Challenge 2021. *PHM Society*, 1-6.

Chao, M. A., Kulkarni, C., Goebel, K., & Fink, O. (2022, 1). Fusing physics-based and deep learning models for prognostics. *Reliability Engineering & System Safety*, *217*, 107961. doi: 10.1016/J.RESS.2021.107961

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In (pp. 785–794). ACM. doi: 10.1145/2939672

Coble, J., Ramuhalli, P., Bond, L., Hines, J. W., & Upadhyaya, B. (2015). A review of prognostics and health management applications in nuclear power plants. *International Journal of Prognostics and Health Management*, *6*(3), 1–22.

DeVol, N., Saldana, C., & Fu, K. (2021). Inception Based Deep Convolutional Neural Network for Remaining Useful Life Estimation of Turbofan Engines. In (Vol. 13). PHM Society. doi: 10.36001/PHMCONF.2021.v13i1.3109

Genuer, R., Poggi, J. M., Tuleau-Malot, C., & Villa-Vialaneix, N. (2017). Random Forests for Big Data. *Big Data Research*, *9*, 28–46. doi: 10.1016/j.bdr.2017.07.003

Gupta, M., Wadhvani, R., & Rasool, A. (2023, 1). A real-time adaptive model for bearing fault classification and remaining useful life estimation using deep neural network. *Knowledge-Based Systems*, *259*, 110070. doi: 10.1016/J.KNOSYS.2022.110070

Innes, M. (2018). Flux: Elegant machine learning with Julia. *Journal of Open Source Software*, *3*. doi: 10.21105/JOSS.00602

Innes, M., Edelman, A., Fischer, K., Rackauckas, C., Saba, E., Shah, V. B., & Tebbutt, W. (2019). *A Differentiable Programming System to Bridge Machine Learning and Scientific Computing.*

Jollife, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philos Trans A Math Phys Eng Sci*, *374*(2065). doi: 10.1098/RSTA.2015.0202

Kong, Z., Jin, X., Xu, Z., & Zhang, B. (2022). Spatio-Temporal Fusion Attention: A Novel Approach for Remaining Useful Life Prediction Based on Graph Neural Network. *IEEE Transactions on Instrumentation and Measurement*, *71*. doi: 10.1109/TIM.2022.3184352

Lai, X., Qiu, T., Shui, H., Ding, D., & Ni, J. (2023, 4). Predicting future production system bottlenecks with a graph neural network approach. *Journal of Manufacturing Systems*, *67*, 201-212. doi: 10.1016/J.JMSY.2023.01.010

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

Li, C., Sanchez, R. V., Zurita, G., Cerrada, M., Cabrera, D., & Vásquez, R. E. (2016). Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. *Mechanical Systems and Signal Processing*, *76–77*, 283–293. doi: 10.1016/J.YMSSP.2016.02.007

Li, T., Zhou, Z., Li, S., Sun, C., Yan, R., & Chen, X. (2022, 4). The emerging graph neural networks for intelligent fault diagnostics and prognostics: A guideline and a benchmark study. *Mechanical Systems and Signal Processing*, *168*, 108653. doi: 10.1016/J.YMSSP.2021.108653

Liao, L., & Köttig, F. (2014). Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Transactions on Reliability*, *63*, 191–207. doi: 10.1109/TR.2014.2299152

Lövberg, A. (2021, 12). Remaining Useful Life Prediction of Aircraft Engines with Variable Length Input Sequences. In (Vol. 13). PHM Society. doi: 10.36001/PHMCONF.2021.V13I1.3108

Mahamad, A. K., Saon, S., & Hiyama, T. (2010). Predicting remaining useful life of rotating machinery based artificial neural network. *Computers and Mathematics with Applications*, *60*(4), 1078–1087. doi: 10.1016/J.CAMWA.2010.03.065

Maier, O., Wilms, M., von der Gablentz, J., Krämer, U. M., Münte, T. F., & Handels, H. (2015). Extra Tree forests for sub-acute ischemic stroke lesion segmentation in MR sequences. *Journal of Neuroscience Methods*, *240*, 89–100. doi: 10.1016/j.jneumeth.2014.11.011

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Édouard Duchesnay (2012). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Santos, P., Maudes, J., & Bustillo, A. (2018). Identifying maximum imbalance in datasets for fault diagnosis of gearboxes. *Journal of Intelligent Manufacturing*, *29*(2), 333–351. doi: 10.1007/S10845-015-1110-0

Senoner, J., Netland, T., & Feuerriegel, S. (2022). Using Explainable Artificial Intelligence to Improve Process Quality: Evidence from Semiconductor Manufacturing. *Management Science*, *68*(8), 5704-5723. doi: 10.1287/mnsc.2021.4190

Shao, H., Jiang, H., Lin, Y., & Li, X. (2018). A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. *Mechanical Systems and Signal Processing*, *102*, 278–297. doi: 10.1016/J.YMSSP.2017.09.026

Solís-Martín, D., Galán-Páez, J., & Borrego-Díaz, J. (2021). A Stacked Deep Convolutional Neural Network to Predict the Remaining Useful Life of a Turbofan Engine. In (Vol. 13). PHM Society. doi: 10.36001/PHMCONF.2021.V13I1.3110

Song, T., Liu, C., Wu, R., Jin, Y., & Jiang, D. (2022, 5). A hierarchical scheme for remaining useful life prediction with long short-term memory networks. *Neurocomputing*, *487*, 22-33. doi: 10.1016/J.NEUCOM.2022.02.032

Tsui, K. L., Chen, N., Zhou, Q., Hai, Y., & Wang, W. (2015). Prognostics and health management: A review on data driven approaches. *Mathematical Problems in Engineering*, *2015*, 793161. doi: 10.1155/2015/793161

Wu, J. Y., Wu, M., Chen, Z., Li, X., & Yan, R. (2021, 1). A joint classification-regression method for multi-stage remaining useful life prediction. *Journal of Manufacturing Systems*, *58*, 109-119. doi: 10.1016/J.JMSY.2020.11.016

Wu, X., & Ye, Q. (2016, 7). Fault diagnosis and prognostic of solid oxide fuel cells. *Journal of Power Sources*, *321*, 47-56. doi: 10.1016/J.JPOWSOUR.2016.04.080