

A Change Point Detection Integrated Remaining Useful Life Estimation Model under Variable Operating Conditions

Anushiya Arunan

Engineering Product Development
Singapore University of Technology and Design
Email: anushiya_arunan@mymail.sutd.edu.sg

Yan Qin

Engineering Product Development
Singapore University of Technology and Design
Email: zdqinyan@gmail.com

Xiaoli Li

Institute for Infocomm Research
Agency for Science, Technology and Research
Email: xlli@i2r.a-star.edu.sg

Chau Yuen

School of Electrical and Electronics Engineering
Nanyang Technological University
Email: chau.yuen@ntu.edu.sg

Abstract—By informing the onset of the degradation process, health status evaluation serves as a significant preliminary step for reliable remaining useful life (RUL) estimation of complex equipment. However, existing works rely on *a priori* knowledge to roughly identify the starting time of degradation, termed the change point, which overlooks individual degradation characteristics of devices working in variable operating conditions. Consequently, reliable RUL estimation for devices under variable operating conditions is challenging as different devices exhibit heterogeneous and frequently changing degradation dynamics. This paper proposes a novel temporal dynamics learning-based model for detecting change points of individual devices, even under variable operating conditions, and utilises the learnt change points to improve the RUL estimation accuracy. During offline model development, the multivariate sensor data are decomposed to learn fused temporal correlation features that are generalisable and representative of normal operation dynamics across multiple operating conditions. Monitoring statistics and control limit thresholds for normal behaviour are dynamically constructed from these learnt temporal features for the unsupervised detection of device-level change points. The detected change points then inform the degradation data labelling for training a long short-term memory (LSTM)-based RUL estimation model. During online monitoring, the temporal correlation dynamics of a query device is monitored for breach of the control limit derived in offline training. If a change point is detected, the device's RUL is estimated with the well-trained offline model for early preventive action. Using C-MAPSS turbofan engines as the case study, the proposed method improved the accuracy by 5.6% and 7.5% for two scenarios with six operating conditions, when compared to existing LSTM-based RUL estimation models that do not consider heterogeneous change points.

Index Terms—Temporal dynamics learning, change point detection, degradation analysis, remaining useful life estimation,

canonical variate analysis, long short-term memory network.

I. INTRODUCTION

Production efficiency and process safety of complex systems are contingent on individual devices operating reliably. Accurate remaining useful life (RUL) estimation is a key enabler of device reliability as condition based, and preventive maintenance can be timely scheduled based on the RUL information. Generally, the RUL of critical assets is defined as the length of time from the current time to end of useful life, i.e., complete failure [1]. An accurate RUL estimation provides crucial guidance for early action to prevent downtime due to unexpected breakdown.

With the advent of Industry 4.0 and the massive amount of data generated from Industrial Internet of Things sensors, research interest in data-driven RUL estimation models has grown rapidly. These models are developed to capitalise on the temporal nature of sensor data, and can be broadly grouped into classical statistics-based models and deep learning based approaches. In statistics-based models, Ordóñez *et al.* [2], for instance, captured time dependence with a combined autoregressive integrated moving average - support vector machine regression model for RUL estimation of engines. Wang *et al.* [3] proposed a continuous hidden Markov model to extract degradation state information from time sequences to feed their RUL estimation model for milling tools. Zheng *et al.* [4] utilised a sliding window approach to input time series sensor data into an extreme learning machine based RUL estimation model. However, some drawbacks of these approaches are

the inability and computational impracticality of accounting for long-term time dependencies such as in Markov models [3], and the lost time dependency information when sliding windows are assumed to be independent of each other [4].

To address these shortcomings, deep learning approaches have gained popularity in recent years as the backbone for fault diagnosis and RUL estimation. These models have been proposed for a range of critical equipment such as machine bearings [5], [6], [7], [8], cooling systems [9], [10], engines [11], [12], [13], [14], [15], and even batteries [16], [17], [18], [19]. Particularly, the long-short term memory (LSTM) architecture is of interest compared to standard recurrent neural networks due to its ability to capture both short-term patterns and long-term dependencies in time series via the information-sieving mechanisms of the LSTM's input, forget, and output gates [20].

Despite the superior abilities of LSTM, existing RUL estimation models often use simplifying assumptions or domain knowledge-reliant literature values for modelling the RUL progression through a machine's lifetime [21], [22], [23], [24], [25]. Generally, critical assets operate normally in the beginning and the onset of degradation only occurs after an uncertain time point, defined as the change point [26]. Taking turbofan engines for instance, this non-linear progression of RUL is often represented in a piecewise manner, where the RUL is capped at a constant upper limit during initial operation cycles, and only starts to decrease after some time in operation. The upper RUL limit is typically capped using prior literature values from operational experiments [27], [21], [22], [23], [24]. However, an obvious shortcoming of such an approach is the considerable domain expertise needed for selecting suitable change points.

There have been a few budding research efforts recognising the need for data-guided change point detection to improve RUL estimation models. For instance, Wu *et al.* [28] utilise a support vector machine-based anomaly detector to identify the change point prior to an LSTM-based RUL estimation. However, their method is evaluated on only a small test size of 20 engines, and the work focuses solely on late-stage RUL estimation (defined as the last 50 cycles before failure). Meanwhile, Shi and Chehade [26] put forward a dual-LSTM model to consider the heterogeneous change points of different devices in their RUL estimation. The first LSTM model classifies if an engine is in a normal or degradation state to detect the change point, while the second LSTM model performs the RUL estimation. Their work similarly focuses on late-stage RUL estimation, and though the RUL estimation performance was promising for cases with single operating conditions, the performance under multiple operating conditions is unknown. Interested readers may also refer to Appendix A for a detailed comparison of these existing works and our current work.

In practice, a device may experience variable and frequently switching operating conditions, resulting in heterogeneous degradation behaviour among different devices. Hence, it is difficult to achieve accurate RUL estimation when individual differences in degradation behaviour are overlooked. There-

fore, in this paper, we discuss and address the following two crucial yet unsolved challenges:

- Variable operating conditions naturally produce disparate degradation processes, resulting in different change points for individual devices. However, current approaches of degradation modelling still apply a prior knowledge-reliant, fixed representation for all devices of the same type. Particularly, existing works have not investigated how specific change points of individual devices can be identified from each device's degradation behaviour, and utilised for enhancing RUL estimation capabilities.
- Health status evaluation of whether a device is in normal operation or degradation state is an essential preliminary step for developing reliable RUL estimation models. However, existing degradation modelling studies require well-labelled data to distinguish between different states. As in-depth knowledge of underlying operating conditions and labelled data are not always available, supervised models can be unreliable when extended to devices with different working principles. Thus, there is an urgent need for an unsupervised, generalizable, and data-driven method to account for dynamic degradation behaviours within RUL estimation models.

To address these gaps thoroughly and handle the challenges of variable operating conditions, we propose in-depth analysis of local temporal dynamics for health status evaluation and change point detection, prior to an LSTM-based RUL estimation model. Here, the local temporal dynamics is defined as the short-term correlations between a limited number of adjacent past and future lags of sensor measurements. Notably, we successfully extract out generalisable temporal variations that are representative of normal operation dynamics across multiple operating conditions, by a novel leveraging of canonical variate analysis (CVA), to automatically detect change points based on significant changes to these temporal variations. Specifically, latent local temporal correlations are learnt and extracted from raw sensor measurements to dynamically construct the monitoring statistics and control limit for the unsupervised detection of device-level change points. The change points then inform the degradation data labelling for training the LSTM-based RUL estimation model, which is now health-status cognizant. Using the trained model, an online query device's change point and RUL can be estimated for early preventive action. Overall, the key contributions of this paper are:

- i) We introduce a novel temporal learning methodology for analysing the latent temporal dynamics of sensor measurements and tracking device degradation progression under variable operating conditions.
- ii) We propose a comprehensive and generalisable health status-dependent RUL estimation model, where the RUL estimation of individual devices is enhanced with precise change point detection. Our unsupervised change point detection method circumvents the need for domain expertise and ground-truth based labelling of train data.

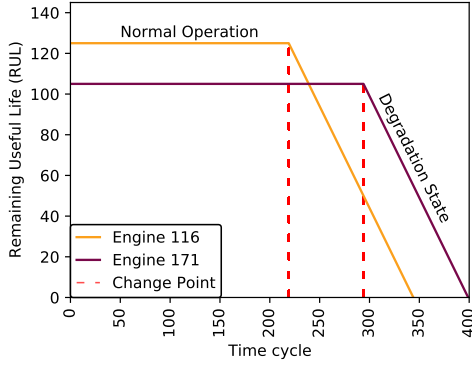


Fig. 1. Piecewise RUL function showing the relationship between the change point and upper RUL limit value, using randomly selected engines of FD004 in the Commercial Modular Aero-Propulsion System Simulation dataset.

- iii) We validate the criticality of accounting for heterogeneous change points of individual devices, especially complex devices with multiple operating conditions, by achieving 5.6% - 7.5% improvement in RUL estimation accuracy.

The remainder of the paper is organised as follows. Section II describes the preliminaries for a better understanding of the proposed method. Section III outlines the proposed method for change point detection and RUL estimation. Section IV discusses the data, experiments, and results. Finally, Section VI concludes and highlights future research directions.

II. PRELIMINARY

This section builds the premise for change point integrated RUL estimation and discusses the concept of change point and the standard LSTM model to build the integrated framework.

A. Definition of Change Point

Heimes [27] is one of the first influential works to popularise the use of a piecewise-linear function (i.e., constant RUL, followed by a linearly decreasing RUL) to model a device's RUL progression throughout its lifespan. The need for representing a device's degradation process in a piecewise manner arises because its lifecycle can be broadly divided into two states: a healthy, normally operating state and a degradation state. During the initial operating cycles before the change point, degradation is often negligible, and it can be reasonably assumed that the RUL remains relatively unchanged (constant) for practical modelling purposes. The device's RUL only starts decreasing distinctively when degradation begins after a yet-to-be determined time point, termed the change point. As seen in Fig. 1, the RUL is capped by an upper limit during normal operation and diminishes only during the degradation state. The change point marks the shift from normal operation to the degradation state. As a side, the RUL is assumed to decrease linearly after the change point in our work, following [27], [21], [22], [23], [24], [29]. However, other variants such as a non-linear RUL decay after the change point can also be easily considered depending on the dataset characteristics and domain application.

B. Change Point Integrated RUL Estimation

The piecewise RUL target label is a key input for RUL estimation models. However, an important difference in the existing approaches of constructing the piecewise labels is that only fixed literature values of the upper RUL limit are available for capping the RUL function, and thus, these values are directly used [21], [22], [23], [24], [25]. Consequently, the change points of individual devices are not explicitly known or investigated. In contrast, a change point integrated RUL estimation seeks to detect the change point first, and then calculate the unique upper RUL limit for each device following:

$$y_j^{max} = k_j^{max} - k_j^{cp} \quad (1)$$

where y_j^{max} is the upper RUL limit for device j , given its change point, k_j^{cp} and its maximum lifespan k_j^{max} .

The piecewise degradation data constructed from the learnt change points is fed to the LSTM model discussed next to form the change point integrated RUL estimation framework.

C. LSTM Model for RUL Estimation

An LSTM-based RUL estimation models the non-linear relationship between input sensor data and the piecewise RUL labels (i.e., degradation data). A basic LSTM unit is a cell with three gates (input, forget and output) to sieve information flow through the cell. LSTM is a recurrent network as both its cell state \mathbf{c}_k and hidden state, \mathbf{h}_k at time k holds the memory from the previous cell state, \mathbf{c}_{k-1} , hidden state \mathbf{h}_{k-1} and input \mathbf{x}_k [20]. The predicted RUL \hat{y}_k is determined by \mathbf{h}_k . For interested readers, further details of the standard LSTM architecture can be found in [20].

III. CHANGE POINT DETECTION INTEGRATED MODEL FOR REMAINING USEFUL LIFE ESTIMATION

The change point detection integrated RUL estimation model can be divided into offline modelling and online monitoring. In offline modelling, a change point detection model first analyses the local temporal dynamics of sensor measurements to learn the start time of degradation, i.e., the change point. Then, the learnt change points are utilised to calculate upper RUL limit values for transforming the RUL labels of a device as a piecewise function. With the transformed labels, an LSTM model is developed and trained. During online monitoring, a query device is monitored for the occurrence of a change point and its RUL is estimated with the offline trained models. The steps of the proposed methodology, from change point detection to RUL estimation, are detailed below, and summarised in Fig. 2.

A. Change Point Detection using Temporal Correlations

Conventionally, CVA is used for finding associations between two multivariate datasets, where the multiple variables in each dataset are considered as a whole. For multiple variables to be considered as a whole, the variables are transformed via appropriate linear combinations (i.e., projections) to a fused latent variable, termed the canonical variate [30]. In this paper, we capitalise CVA's association finding ability

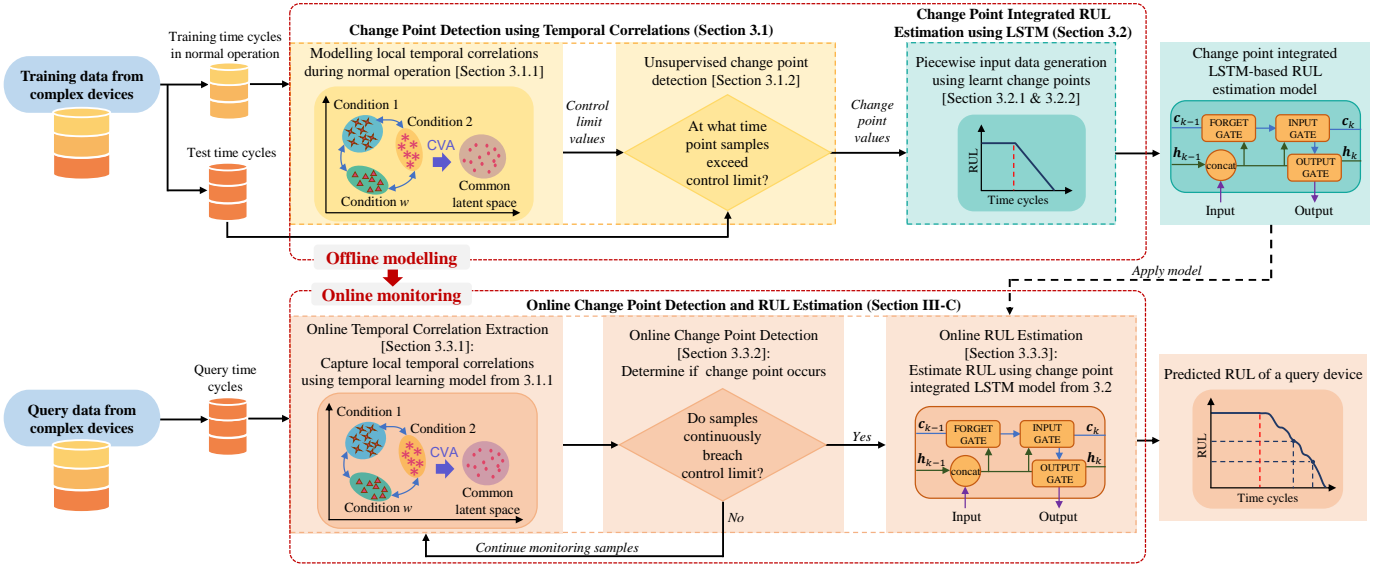


Fig. 2. Overall structure of proposed change point detection-based RUL estimation method.

in an innovative way to create an in-depth temporal dynamics monitoring model for detecting degradation change points.

1) *Modelling of Local Temporal Correlations during Normal Operation*: The procedure for building and training the CVA-based temporal learning model are outlined next. A training set of sensor data during normal operation is required to calculate the canonical variate matrices, health monitoring statistics, and a Control Limit (CL), which acts as an upper threshold for the monitoring statistics. The monitoring statistics of any new test data can be compared against the CL value previously calculated with the normal operation training data to detect statistically significant breaches above the CL, and consequently, the change point at which degradation begins.

To prepare the training data of each device for capturing the significant latent temporal correlations amongst the sensor measurements, we start with the feature matrix of sensor data obtained during normal operation, $\mathbf{X} \in \mathbb{R}^{m \times N}$, consisting of m sensor variables (e.g., temperature, pressure) and N time series observations. To relate the temporal correlations of multiple sensor data, the past and future vectors, $\mathbf{x}_{p,k}$ and $\mathbf{x}_{f,k}$ are formed by expanding each sensor measurement at time cycle, k by p past lags and f future lags:

$$\begin{aligned} \mathbf{x}_{p,k} &= [\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \dots, \mathbf{x}_{k-p}]^T \in \mathbb{R}^{mp} \\ \mathbf{x}_{f,k} &= [\mathbf{x}_k, \mathbf{x}_{k+1}, \dots, \mathbf{x}_{k+f-1}]^T \in \mathbb{R}^{mf} \end{aligned} \quad (2)$$

where p and f represent the number of lags of sensor measurements used for the temporal correlation modelling.

The final step of training data construction is concatenating $\mathbf{x}_{p,k}$ and $\mathbf{x}_{f,k}$ vectors in the variable-wise direction to form the comprehensive past and future matrices, \mathbf{X}_p and \mathbf{X}_f :

$$\begin{aligned} \mathbf{X}_p &= [\mathbf{x}_{p,p+1}, \mathbf{x}_{p,p+2}, \dots, \mathbf{x}_{p,p+\tilde{N}}] \in \mathbb{R}^{mp \times \tilde{N}} \\ \mathbf{X}_f &= [\mathbf{x}_{f,p+1}, \mathbf{x}_{f,p+2}, \dots, \mathbf{x}_{f,p+\tilde{N}}] \in \mathbb{R}^{mf \times \tilde{N}} \end{aligned} \quad (3)$$

where $p = f$ and $\tilde{N} = N - f - p + 1$ for N observations.

The data in \mathbf{X}_p and \mathbf{X}_f are standardised to zero mean and unit variance with respect to each sensor variable to prevent large values from skewing subsequent calculations. The transformation matrices required to calculate the canonical variates are derived from the singular value decomposition (SVD):

$$\Sigma_{ff}^{-1/2} \Sigma_{fp} \Sigma_{pp}^{-1/2} = \mathbf{U} \mathbf{D} \mathbf{V}^T \in \mathbb{R}^{mf \times mp} \quad (4)$$

where $\Sigma_{ff}^{-1/2}$ and $\Sigma_{pp}^{-1/2}$ are the covariance matrices of \mathbf{X}_f and \mathbf{X}_p respectively, and Σ_{fp} is the cross-covariance matrix of \mathbf{X}_f and \mathbf{X}_p ; \mathbf{U} and \mathbf{V} are the left and right singular matrices respectively, and \mathbf{D} is a diagonal matrix of non-negative singular values.

The system canonical variates, i.e., the dominant variates with large correlations, \mathbf{Z} and the residual variates with low correlations, \mathbf{E} are calculated as the linear combinations of \mathbf{X}_p by applying the respective transformation matrices, $\mathbf{V}_r^T \Sigma_{pp}^{-1/2}$ and $(\mathbf{I} - \mathbf{V}_r \mathbf{V}_r^T) \Sigma_{pp}^{-1/2}$ to \mathbf{X}_p :

$$\mathbf{Z} = \mathbf{V}_r^T \Sigma_{pp}^{-1/2} \mathbf{X}_p \in \mathbb{R}^{r \times \tilde{N}} \quad (5)$$

$$\mathbf{E} = (\mathbf{I} - \mathbf{V}_r \mathbf{V}_r^T) \Sigma_{pp}^{-1/2} \mathbf{X}_p \in \mathbb{R}^{mp \times \tilde{N}} \quad (6)$$

where r is the number of system canonical variates.

The changes in canonical variates at each time cycle k is quantitatively measured by two health monitoring statistics, which are the Hotelling T^2 statistic and Squared Prediction Error Q statistic. The T^2 and Q statistics are complementary as T^2 captures the total variation of the system canonical variates, while the Q statistic measures the variations of errors in the residual space:

$$T_k^2 = \sum_{i=1}^r z_{i,k}^2 \quad (7)$$

$$Q_k = \sum_{i=1}^{mp} \varepsilon_{i,k}^2 \quad (8)$$

Algorithm 1 Unsupervised change point detection

Input: Test data, \mathbf{X}^{test} from time, τ till end of life, k^{max}
Control limits from normal operation, CL_{T^2} and CL_Q
Output: Change points, $k_{T^2}^{cp}$ and k_Q^{cp}
Initialise sampling time, $k = \tau$
1: Construct \mathbf{X}_p^{test} using Eq. (3)
2: Determine \mathbf{Z}^{test} and \mathbf{E}^{test} using Eqs. (9) and (10)
3: **for** $k = \tau$ to k^{max} **do**
4: Calculate $T_{k_test}^2$ and Q_{k_test} using Eqs. (11) and (12)
5: **end for**
6: Determine change point, $k_{T^2}^{cp}$ based on $T_{k_test}^2$:
7: Initialise $k = \tau$
8: **while** ($k \leq k^{max}$) **do**
9: **if** $T_{k_test}^2(k) \geq CL_{T^2}$ for all k till k^{max} **then**
10: **return** $k = k_{T^2}^{cp}$
11: **break**
12: **else**
13: $k = k + 1$
14: **end if**
15: **end while**
16: Repeat steps (7) through (15) to determine change point, k_Q^{cp} based on Q_{k_test} using corresponding control limit CL_Q

where $z_{i,k}$ and $\varepsilon_{i,k}$ are the elements in row i and column k of the respective canonical variate matrices \mathbf{Z} and \mathbf{E} .

Lastly, the statistically significant CL values for the health monitoring statistics are calculated. For instance, at a statistical significance level of $\alpha = 0.99$, the CL establishes an upper threshold value, below which 99% of T^2 and Q statistic sample values will fall during normal operation. To calculate the CL of T^2 and Q , their probability density functions has to be established. As non-linear systems may not follow a Gaussian distribution, the probability distribution of T^2 and Q is modelled using Kernel Density Estimation following [30]. The CL values for T^2 and Q , defined as CL_{T^2} and CL_Q , are then obtained by solving $P(T^2 < CL_{T^2}) = \alpha$ and $P(Q < CL_Q) = \alpha$, respectively.

2) *Unsupervised Change Point Detection:* With the CL values calculated using training data from normal operation, we can assess the remaining sensor data $\mathbf{X}^{test} \in \mathbb{R}^{m \times N}$ of a device, containing time series sensor data from normal operation and degradation states, to identify its change point. The change point detection strategy is summarised in Algorithm 1, and the detailed steps are discussed henceforth.

First, the lagged past matrix \mathbf{X}_p^{test} is constructed following Eq. (3). Next, similar to the approach for training data in Eqs. (5) through (8), the canonical variates and monitoring statistics T^2 and Q are calculated for the test data as follows:

$$\mathbf{Z}^{test} = \mathbf{V}_r^T \Sigma_{pp}^{-1/2} \mathbf{X}_p^{test} \in \mathbb{R}^{r \times \tilde{N}} \quad (9)$$

$$\mathbf{E}^{test} = (\mathbf{I} - \mathbf{V}_r \mathbf{V}_r^T) \Sigma_{pp}^{-1/2} \mathbf{X}_p^{test} \in \mathbb{R}^{mp \times \tilde{N}} \quad (10)$$

$$T_{k_test}^2 = \sum_{i=1}^r z_{i,k_test}^2 \quad (11)$$

$$Q_{k_test} = \sum_{i=1}^{mp} \varepsilon_{i,k_test}^2 \quad (12)$$

where z_{i,k_test} and ε_{i,k_test} are elements in row i and column k_test of the respective matrices \mathbf{Z}^{test} and \mathbf{E}^{test} .

The monitoring statistics T^2 and Q of the test data are compared against the control limits CL_{T^2} and CL_Q calculated previously during normal operation. Monitoring statistics values below the control limit indicate normal operations, while a continuous breach above the CL threshold indicates some fault development and a shift into a degradation state. In practice, there is typically a time period, where the monitoring statistics fluctuate above and below the CL threshold before the engine actually starts to degrade. We name this period the “transition period”. Thus, to ensure that an accurate onset of degradation that is past the transition period is captured, the change point, k^{cp} is defined as the first time point at which a continuous and consistent breach above the CL starts. The change points are deemed to occur when either the T^2 or Q monitoring statistics continuously exceed their respective CL values, and they are determined by solving for $k_{T^2}^{cp}$ and k_Q^{cp} :

$$T_{k_test}^2(k_{T^2}^{cp}) \geq CL_{T^2} \quad \forall \quad k \in [k_{T^2}^{cp}, k^{max}] \quad (13)$$

$$Q_{k_test}(k_Q^{cp}) \geq CL_Q \quad \forall \quad k \in [k_Q^{cp}, k^{max}] \quad (14)$$

where $k_{T^2}^{cp}$ and k_Q^{cp} are the change points based on T^2 and Q statistics respectively, and k^{max} is the device lifespan.

B. Change Point Integrated RUL Estimation using LSTM

This section discusses how the learnt change points from Section III-A2 are holistically utilised to enhance the quality of the input data used for training our change point integrated RUL estimation model. The input data consists of two interconnected components, the sensor features and the ground truth RUL labels, which have to be considered concurrently when accounting for the change points within the modelling process. We detail this change point-informed generation of the RUL labels and pre-processing of sensor features in the two subsequent sections.

1) *Change Point-informed RUL Label Generation:* To generate the RUL labels, we take inspiration from Heimes’s seminal experiments [27] to adopt a piecewise-linear model (i.e., constant RUL prior to the start of degradation, followed by a linearly decreasing RUL till complete failure). In most existing papers on turbofan engines for instance, a constant value of 130 [21], [11] is recommended as the upper RUL limit for all devices, even under variable operating conditions. In contrast, learning the change point of individual devices allows for the unique RUL upper limit to be determined for each device according to Eq. (1). Fig. 1 illustrates the relationship between the change point and the upper RUL limit. For example, Engine 116 of the FD004 dataset in the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) turbofan engine degradation dataset [31] has a maximum lifespan of 344 cycles and a change point at around 240 cycles. Therefore, its upper RUL limit will be 104.

After the change point, the choice of a linear or non-linear model for the RUL labelling depends on the equipment type and the characteristics of the generated data. In our work, we assume a linear decay of the RUL after the change point based on the well-studied suitability [21], [22], [23], [24] of

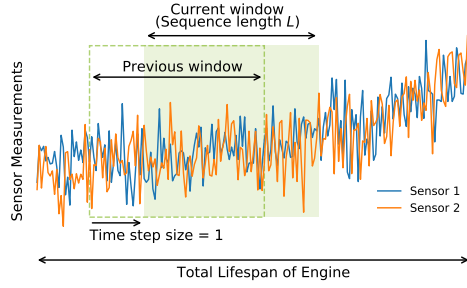


Fig. 3. Data segmentation using sliding window method with two randomly selected sensor signals shown as examples.

this assumption for our downstream case study. However, it should be noted that our data-driven change point detection methodology is designed to detect departures from normal operation behaviour solely from the sensor data's temporal dynamics. It does not assume any underlying model for the RUL decrease after the change point, and thus, it can be easily applied to other domains and datasets experiencing a non-linear RUL decay after the change point.

2) Change Point-informed Sensor Data Standardisation:

Feature scaling is a standard data pre-processing procedure required before inputting features to an LSTM model. This prevents variables with relatively large values from dominating and skewing model training and convergence. A popular standardisation technique is Z-score normalisation [21], where each sensor variable is scaled to have zero mean and unit variance.

In our work, we enhance the Z-score normalisation process with the learnt change points to perform a piecewise standardisation of the sensor data. First, the mean and standard deviation of sensor data before the change point (i.e., data in normal operation) are calculated, and these values are used to standardise the entire sensor data of both train and test devices. This normal operation based standardisation allows the sensor data variations experienced during the degradation state to be better contrasted and amplified against normal operation variations.

The aforementioned change point integrated feature data is then segmented into smaller sequences of length L using a sliding window approach, as shown in Fig. 3. The window is shifted through the entire feature data by a step size of 1 at a time. This generates smaller sequence segments, which are consecutively fed into an LSTM network, together with the piecewise-constructed RUL labels, to model the relationship between the feature data and the RUL. The proposed LSTM model consists of 3 stacked layers to increase its capacity to learn important, latent dependencies in the change point-informed input data that can aid accurate RUL estimation. The model is trained to minimise the loss function, the mean squared error of the predicted RUL and true RUL labels, and subsequently, the trained model can be utilised to predict the RUL of any new query device.

C. Online Change Point Detection and RUL Estimation

With the well-trained change point detection and RUL estimation models, any query sample \mathbf{X}^{query} can be monitored for change point detection and RUL estimation. The online monitoring scheme is summarised stepwise below.

1) *Online Temporal Correlation Extraction*: Given a query sample $\mathbf{X}^{query} \in \mathbb{R}^{m \times N}$ from a device, its past lagged matrix is constructed using Eq. (3). Next, the local temporal correlation features are extracted for the calculation of T_k^2 and Q_k monitoring statistics following Eqs. (9) to (12).

2) *Online Change Point Detection*: The calculated T_k^2 and Q_k statistics are checked against their respective control limits, CL_{T^2} and CL_Q , established in Section III-A1, to assess if the queried sample falls under normal operation. If the T_k^2 and Q_k statistics remain below the CL, the device is operating normally. It is continued to be monitored with no further action. In contrast, if the T_k^2 or Q_k statistics continuously exceed the CL, the device is no longer operating normally. To safely conclude this shift from normal operation to degradation state, the number of time cycles λ that the T_k^2 or Q_k statistics need to continuously breach the CL should be at least as high as the maximum number of consecutive time cycles the breach occurs for during normal operation and, in any transition period to the degradation state:

$$\lambda = \arg \max_{\Delta k} \begin{cases} T_k^2 \geq CL_{T^2} \quad \forall k \in [k, k + \Delta k] \\ Q_k \geq CL_Q \quad \forall k \in [k, k + \Delta k] \end{cases} \quad (15)$$

where Δk is the largest increment in time period that the CL breach is sustained for. Following the breach, the change points $k_{T^2}^{cp}$ and k_Q^{cp} are detected according to Eqs. (13) and (14).

On a related note, there could be other domain applications in practice, where detection of multiple change points is desired. In such cases, the value of Δk can be appropriately tweaked based on observed normal operation behaviour to fine-tune the sensitivity of the system to detect multiple change points.

3) *Online RUL Estimation*: If a change point is detected, this indicates the onset of degradation, and an RUL estimation is necessary for planning preventive maintenance. The feature data of the queried sample is processed by piecewise standardisation and sliding window based sequence segmentation before it is fed into the LSTM-based RUL estimation model.

At this juncture, it should also be highlighted that, in industrial processes, data drifts (i.e., changes in data distribution of incoming query data) may gradually occur over time due to factors such as replacement of ageing equipment or changes in operational procedures. As the change point detection model is designed to generalise well over variable operating conditions, we expect our change point integrated RUL estimation model to be reasonably robust against minor data drifts in the near-term.

However, over the long term, substantial data drifts could occur, thus, it is prudent to establish a data and model monitoring pipeline. Incoming test data should be periodically assessed for possible changes in data distributions through standard statistical approaches (e.g., Kolmogorov-Smirnov test

[32], Kullback Leibler divergence [33], etc.). If substantial data drifts occur, the CLs may no longer be representative of the new normal operating conditions. This can be easily remedied offline by retraining or updating the original model with the newly available data to learn the latest normal operating conditions or new fault patterns.

IV. EXPERIMENTS AND RESULTS

This section assesses the change point detection and RUL estimation performance of the proposed temporal learning model, using the benchmark C-MAPSS turbofan engine degradation dataset [31]. For a fair comparison, performance evaluation is carried out by comparing against LSTM-based deep learning models [21], [22], [23], [24], [25].

The benchmark C-MAPSS turbofan engine degradation dataset consists of 4 sub-datasets, FD001 to FD004. Each sub-dataset has a varying number of engines, fault modes, and operating conditions. Within each sub-dataset, there is a further division into train and test engines. The dataset is summarised in Table I. FD001 is the simplest sub-dataset with engines experiencing 1 operating condition and 1 fault mode, while FD004 is the most complex with 6 operating conditions and 2 fault modes. In the dataset, there are 21 sensor variables (e.g., temperature, pressure) recorded for each operational time cycle of the engine. For the train engines, the time series sensor data are collected from normal operation until system failure. For the test engines, the data are available up to only some random time before failure. The test engines are used to predict the RUL, i.e., number of remaining operational cycles before failure. As the train engine dataset is a time series of operational cycles from normal operation until failure, the maximum number of operational cycles (lifespan) of each engine can be deduced. For instance, in FD001, the lifespan of engines ranged from 128 to 362 cycles.

A. Change Point Detection using Temporal Correlations

Existing benchmark studies on turbofan engine degradation have extensively discussed which sensors of the C-MAPSS dataset should be selected as features for model training. According to these literature [22], [28], [26], only sensor signals with either increasing or decreasing trends should be selected for model development. Sensors that exhibit erratic trends or remain constant over time do not provide useful information about the degradation process and, therefore, can be excluded as features. To guide our sensor selection process, we employ knowledge from existing literature and our own testing with exploratory, trend-checking plots of the sensor readings. Among the four datasets, FD001 and FD003's sensor signal patterns are similar because these datasets only contain engines experiencing a single operating condition. For FD001 and FD003, sensors 1, 5, 6, 10, 16, 18, and 19 are excluded as features because these sensor readings largely remain constant with zero variation, as shown in the representative sensor signal plots of Fig. 4. On the other hand, datasets FD002 and FD004 consist of engines experiencing multiple operating conditions. Hence, their sensor signal patterns, as reasonably

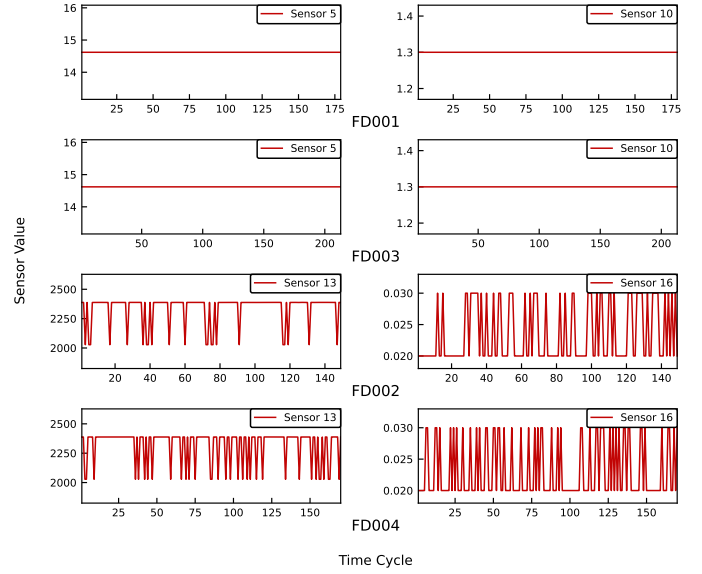


Fig. 4. Sample of uninformative sensor readings with constant values for randomly selected engines from FD001 and FD003, and with erratic, range-bound patterns for randomly selected engines from FD002 and FD004.

expected, differ from that of FD001 and FD003. For FD002 and FD004, sensors 10, 13, 16, 18, and 19 are dropped from being features because they have erratic, range-bound measurements with no obvious increasing or decreasing trends, as shown in the representative sensor signal plots of Fig. 4.

1) Modelling of Local Temporal Correlations during Normal Operation: To recap, the change point detection model has to be first trained on normal operation data to be able to detect statistically significant deviations from normal operation and degradation change points for new test samples that have both normal operation and degradation conditions.

For model training, the train engine dataset is used. As this dataset, naturally, does not label sensor data by whether it is from normal operation or not, some reasonable assumptions are needed. First, engines with relatively lengthy lifespan of at least 200 operational cycles are selected to ensure there is sufficient time series data for the training, validation, and testing phases. Interested readers may refer to Appendix B for details leading to the choice of 200 cycles as the minimum lifespan of train engines. This subset of engines still represents a sizeable dataset for the development of a robust change point detection model as nearly 50% or more of the train engines have a lifespan of at least 200 cycles (refer to the distribution of train engine lifespan in Table I). For the remaining engines with a lifespan of less than 200 cycles, whose change points are not determined by the detection model due to data size insufficiency, we adopt the commonly used literature value of 130 cycles [21] as the RUL upper limit for the piecewise modelling of RUL target labels in the later Section IV-B1.

For the engines with a lifespan of at least 200 cycles, sensor data from first 60 operational cycles of each engine are assumed to be from normal operation. The lagged sensor data, using p past lags and f future lags, are computed using Eqs. (2) and (3). Typically, p and f have the same values and their

TABLE I
DESCRIPTIVE SUMMARY OF C-MAPSS DATASET.

Dataset	FD001	FD002	FD003	FD004
Operating conditions	1	6	1	6
No. of train engines	100	260	100	249
No. of test engines	100	259	100	248
Fault components	High pressure compressor	High pressure compressor	High pressure compressor and fan	High pressure compressor and fan

Distribution of train engine lifespan				
	High pressure compressor	High pressure compressor	High pressure compressor and fan	High pressure compressor and fan

optimal values, i.e., statistically significant number of time lags are determined by comparing the autocorrelation function of the summed squares of the measurements in the training data against a certain confidence interval[34]. However, due to the limited size of normal operation data available in the C-MAPSS dataset, a smaller value of 2 is used for the number of p and f lags in our construction of past and future matrices. Hence, the temporal correlations extracted are termed local. The lagged sensor data matrices are transformed via Eqs. (5) through (8) to calculate the monitoring statistics, T^2 and Q . The optimal number of system canonical variates r to calculate T^2 and Q is determined based on the downstream RUL estimation performance, and it is discussed in greater detail in Section IV-B4. For FD001 to FD003, $r = 15$ resulted in the best RUL estimation performance, whereas, for FD004, which has multiple fault modes and operating conditions, $r = 21$ yielded the best RUL estimation.

The CL of the T^2 and Q statistics is calculated based on a 99% confidence interval. When the monitoring statistics are consistently above the CL for normal operation, we deduce that degradation has begun. To ensure that the first 60 cycles indeed fall under normal operation, the next 20 cycles are taken as validation data and monitored against the previously calculated CL threshold. Using Engine 116 as an example from the most complex FD004 dataset, Figs. 5(a) and 5(b) show that the T^2 and Q statistics for training and validation operational cycles are mostly below the CL with a 99% confidence bound, which is expected from operating data in normal operation. This pattern of the T^2 and Q statistics falling within the 99% CL during validation cycles was observed for all train engines studied as well, leading to the reasonable conclusion that the first 80 cycles represent a state of normal operation.

2) *Unsupervised Change Point Detection*: Remaining operational cycles after the first 80 cycles (60 training and 20 validation cycles) are used as testing cycles containing a yet to be determined change point at which the engine begins to degrade. Fig. 6 plots the testing cycles of three engines selected from FD002 to FD004, which have multiple operating conditions and/or multiple fault modes. Engines of similar lifespan (341 to 344 cycles) were chosen for a fair comparison. Interestingly, the monitoring statistics T^2 and Q fluctuate about the CL threshold for some time, previously termed the

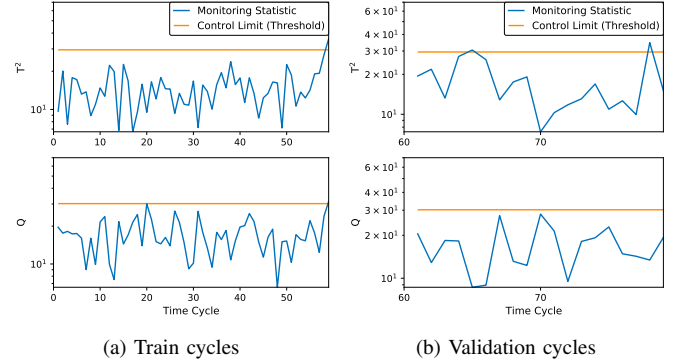
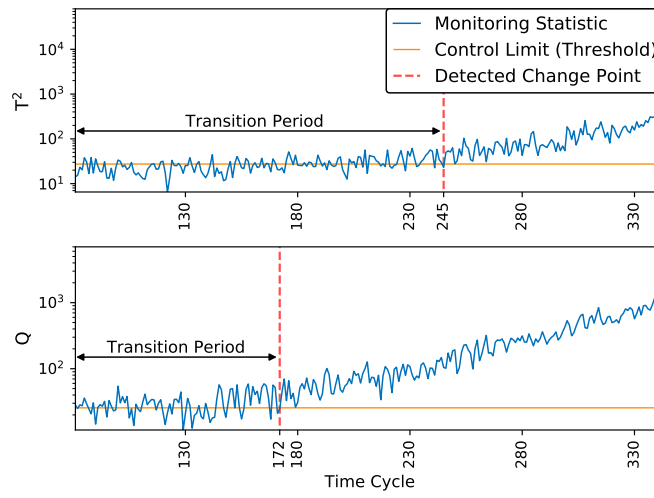


Fig. 5. Monitoring statistics, T^2 and Q during normal operation with six operating conditions (a) first 60 operational cycles, and (b) next 20 operational cycles for Engine 116 of FD004.

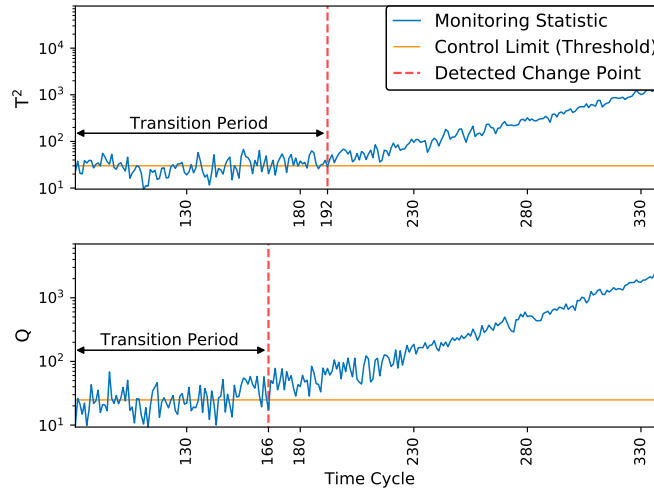
transition period in Section III-A2, before breaching the CL and increasing steadily away from it.

The change point is the latest time cycle at which the T^2 and Q statistics permanently breach the CL and stay above it. The T^2 and Q statistics each yield a change point. The earlier change point of the two values is used to inform the subsequent labelling of the piecewise RUL function for the LSTM-based RUL estimation. Taking Engine 116 of FD004 in Fig. 6(c) as an example, the earlier change point, 240 is selected over the later one, 249. It is reasonable to choose the earlier change point as early warning is preferred in practice to take timely preventive maintenance. It is also worth noting from Fig. 6 that although the three engines selected have similar lifespan, their change points detected are appreciably different due to the varying operating conditions and fault types.

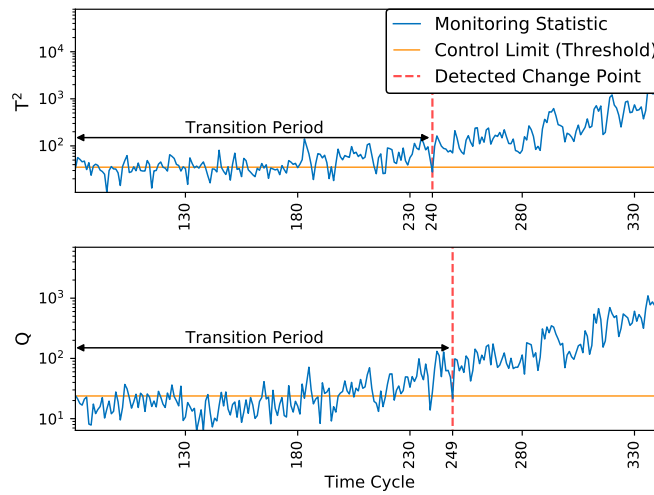
The calculated change points of each engine in the FD002 and FD004 datasets are plotted against the backdrop of their lifespans in Fig. 7. As expected, the change point occurs at later cycles (indicated by larger change point values), for engines with longer lifespans. For engines with a lifespan less than 200 cycles, their change points are monotonically decreasing with respect to their lifespans, because we use a fixed upper RUL limit of 130 cycles as explained earlier.



(a) Engine 118 from FD002



(b) Engine 86 from FD003



(c) Engine 116 from FD004

Fig. 6. Monitoring statistics, T^2 and Q during remaining operational cycles till end of life (test cycles) for (a) Engine 118 of FD002, (b) Engine 86 of FD003, and (c) Engine 116 of FD004.

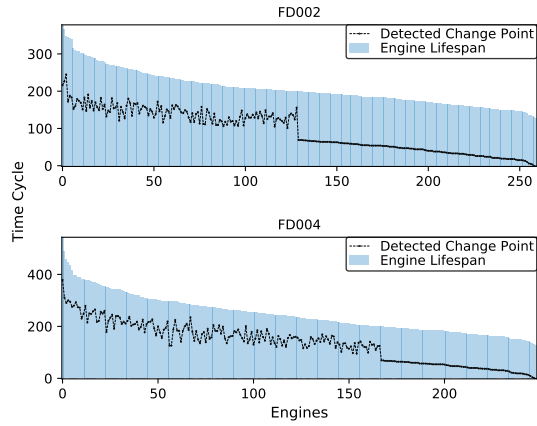


Fig. 7. Detected change point of engines against engine lifespan.

B. Change Point Integrated RUL Estimation using LSTM

1) *Data Preparation:* Before the online RUL estimation, the change point-informed data processing steps discussed in Sections III-B1 and III-B2 are performed. First, RUL labels of the train engines are processed as a piecewise function.

The test engine dataset forms the query devices used for online RUL estimation and performance evaluation. Since the train engine RUL labels were processed in a piecewise manner, the test engine labels are similarly capped with an upper limit. As the C-MAPSS test engine dataset is a static snapshot of sensor data up till a random time before failure, it is not possible to determine change points for the test engines using the online model in Section III-C as it would be the case for a live and continuous stream of query samples. Thus, the literature value of 130 cycles is adopted for the upper RUL limit for the true and estimated RUL of the test engines following [11]. Finally, the sensor data of both train and test engines are scaled by the change point-informed piecewise standardisation described in Section III-B2.

2) *RUL Estimation:* For the LSTM network for RUL estimation, there are already a large number of papers discussing the best architectures for the C-MAPSS dataset. We use the wealth of information available as a start, and verify it with our own testing to construct the our LSTM network with optimised parameters. There are several key hyperparameters that affect the RUL estimation performance. Table II summarises the search space considered for the hyperparameter values and the selected hyperparameter configurations for FD001 to FD004. We discuss some of the notable hyperparameters below.

First, the input sequence length (i.e., the maximum time steps fed to the LSTM cell) is an important factor affecting the learning of temporal dependencies, and consequently, the RUL estimation performance. While longer sequences can provide more contextual information to the model, there is also a risk of model overfitting if sequences are too complex relative to the available training data size [35]. Furthermore, the optimal sequence length is often specific to the dataset and learning task [35], [21]. Thus, we assessed a reasonable range of candidate sequence length $L \in \{30, 40, 50\}$ and $L = 50$ yielded the best RUL estimation for FD001 to FD004.

The next set of hyperparameters, the number of stacked LSTM layers and the number of hidden neurons in a layer, define the LSTM network. Generally, as the number of LSTM layers (i.e., depth of the network) increases, the model's ability to learn more complex, latent relationships between feature variables increases, and thus, the RUL estimation performance may increase if it depends on these relationships. However, adding layers beyond a certain point eventually erodes model performance due to issues such as model overfitting and vanishing gradients in the backpropagation process [36]. For our work, a 3-layer LSTM model yielded the best RUL estimation performance. We discuss the impact of the number of layers on the RUL estimation performance as a sensitivity analysis later in Section IV-B4, and focus here on the role of the optimiser in combating challenges such as vanishing gradients, appropriate learning rates, and slow convergence in deep networks. We assessed both RMSProp [37] and Adam [38] as optimisers. In our experiments, RMSProp, with its ability to adaptively tune the learning rates for the model parameters based on historical gradient information, was found to yield better RUL estimation performance and faster convergence (in 30 epochs).

For the number of hidden neurons, Bengio *et al.* [39] recommends an overcomplete first hidden layer (i.e., a size larger than the input vector dimension) for better generalisability and model performance. Thus, we start off with 256 hidden neurons in the first layer, consistent with [28] and try different configurations for the remaining layers as shown in Table II. Nonetheless, as the number of model parameters increases with deep layers, model overfitting becomes a concern. Thus, we add dropout [40] layers in between the LSTM layers as an important regularisation technique to prevent model overfitting and enhance its generalisability. The dropout ratio specifies the portion of hidden neurons to be randomly dropped out (i.e., excluded) during the training process, thus, inducing the remaining neurons to learn the needed representations for the predictions independent of the randomly dropped neurons. As seen in Table II, candidate dropout ratios considered were $\{0, 0.1, 0.2\}$, where 0 denotes no dropout. The models for all datasets from FD001 to FD004 benefited from the inclusion of dropout, corroborating the crucial role of regularisation in deep networks. For example, the LSTM network for FD001 performed best with a dropout ratio of 0.2 for the first layer and 0.1 for the second layer.

To complete the RUL estimation model, the LSTM layers are combined with a fully connected output layer that decodes the learnt feature representations into a predicted RUL value. The model was trained for 30 epochs as it was sufficient to achieve good convergence and prediction performance on test engines. The performance of the RUL estimation model is evaluated based on two commonly used benchmark metrics, Root Mean Square Error (*RMSE*) and, the Score Function (*SF*) [31]. The *SF* is asymmetric and gives a larger penalty for overestimating the RUL as this can lead to delayed maintenance or even system failure.

3) *RUL Estimation Performance:* In this section, we evaluate the RUL estimation performance of our change point in-

TABLE II
HYPERPARAMETER SELECTION FOR LSTM-BASED RUL ESTIMATION MODEL.

Hyperparameters	Search space	FD001	FD002	FD003	FD004
Sequence length	{30, 40, 50}	50	50	50	50
LSTM layers	{1, 2, 3}	3	3	3	3
Hidden neurons	{32, 64, 100, 128, 256}	(256,128,32)	(256,128,32)	(256,100,32)	(256,100,32)
Dropout ratio	{0, 0.1, 0.2}	(0.2, 0.1)	(0.1, 0.1)	(0.2, 0.1)	(0.1, 0.1)
Learning rate	{0.01, 0.001}	0.001	0.001	0.001	0.001
Optimiser	RMSProp, Adam	RMSProp	RMSProp	RMSProp	RMSProp

TABLE III
PERFORMANCE COMPARISON BETWEEN PROPOSED METHOD AND EXISTING ALGORITHMS (BEST IN **BOLD**, SECOND-BEST UNDERLINED).

Type	Method	FD001		FD002		FD003		FD004	
		RMSE	SF	RMSE	SF	RMSE	SF	RMSE	SF
Conventional Regressors	RF [14]	17.91	479.75	29.59	70456.86	20.27	711.13	31.12	46567.63
	LASSO[14]	19.74	653.85	37.13	276923.89	21.38	1058.36	40.70	125297.19
	XGBoost[41], [42]	15.26	343.60	NA ²	NA ²	19.33	943.76	NA ²	NA ²
LSTM-based Deep Learning	LSTM [21]	16.14	338.00	24.49	4450.00	16.18	852.00	28.17	5500.00
	A-LSTM [22]	14.53	322.44	NA ²	NA ²	NA ²	NA ²	27.08	5649.14
	Bi-LSTM [23]	NA ²	NA ²	25.11	4793.00	NA ²	NA ²	26.61	4971.00
	BS-LSTM [24]	14.89	481.10	26.86	7982.00	15.11	493.40	27.11	5200.00
	CNN-LSTM [25]	14.40	290.00	27.23	9869.00	14.32	316.00	26.69	6594.00
	MC-LSTM[11]	13.71	315.00	NA ²	NA ²	NA ²	NA ²	23.81	4826.00
	Cap-LSTM[43]	12.27	<u>260.00</u>	17.79	<u>1850.00</u>	<u>12.55</u>	<u>217.00</u>	22.05	4570.00
	Att-LSTM [44]	13.95	320.00	<u>17.65</u>	2102.00	12.72	223.00	<u>20.21</u>	3100.00
	GA-CNN-LSTM [45]	15.92	NA ²	22.87	NA ²	17.26	NA ²	26.32	NA ²
	GM-LSTM [46]	14.08	308.00	18.59	1880.00	12.15	221.00	20.91	2633.00
	<i>ChangePoint-LSTM (Ours)</i>	<u>13.59</u>	224.88	16.67	947.99	12.94	207.10	18.69	1360.34
Improvement ¹		-	13.51%	5.55%	48.76%	-	4.56%	7.52%	48.33%

¹ The improvement calculated by comparing proposed model performance against the best-performing benchmark.

² NA is short for not applicable as the results are not provided in cited paper.

tegrated model against an extensive set of benchmarks. These benchmarks range from vanilla LSTM[21] to its state-of-the-art variations[11], [43], but, all still employ fixed literature values for the piecewise RUL construction. The results, in terms of *RMSE* and *SF* metrics, and the percentage improvement over the best-performing benchmarks are presented in Table III. Given the standardisation of the comparison to solely LSTM-based deep learning models, the analysis can also be likened to an ablation study, with results highlighting the impact of accounting for heterogeneous change points in RUL estimation.

As seen from Table III, our model's performance is extremely competitive, even against recent state-of-the-art variations of LSTM. Although our model only utilises a vanilla LSTM architecture, it consistently outperforms other advanced LSTM benchmarks, in terms of *SF*, a metric of greater practical significance due to its higher penalty for overestimating the RUL. Furthermore, our model's notable outperformance, in terms of both *RMSE* and *SF*, for the more complex FD002 and FD004 datasets suggests that factoring in individual change points before RUL estimation is especially important for devices working under variable operating conditions. For these devices, utilising fixed literature values for the upper RUL limit may be inadequate in capturing the complex degradation processes occurring under variable operating conditions. Instead, our results suggest that accounting for heterogeneous starting points of degradation is crucial for achieving accurate and reliable RUL estimation.

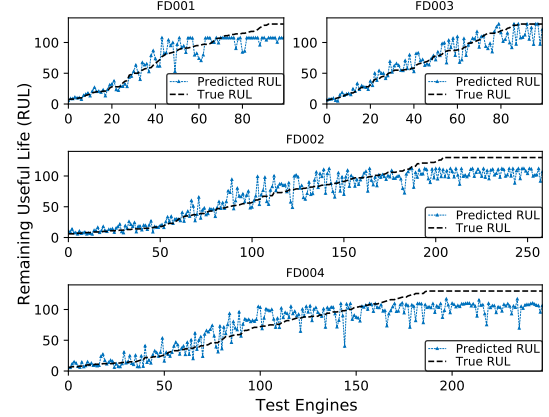


Fig. 8. Comparison between predicted RUL and the ground truth RUL.

Fig. 8 plots the predicted RUL against the true RUL of each engine for an in-depth look into the prediction performance. Particularly, for FD003, the RUL prediction does well for both engines in early operation cycles (true RUL values are large) and engines in later operation cycles (true RUL values are small). For the more complex FD002 and FD004, the RUL prediction is better for engines in later operation cycles. For engines in earlier operation cycles, the predicted RUL tends to be conservatively less than the actual RUL. Nonetheless, the results are very promising given the engines are operating in a complex situation of multiple operating conditions.

4) *Sensitivity Analysis*: The RUL estimation performance is influenced by the quality of the detected change points and the LSTM network's capacity to learn complex feature relationships. In this section, we examine the impact of two key hyperparameters on the RUL estimation performance: the number of system canonical variates r (which indirectly determines the change point) and the number of LSTM layers (an indicator of the depth and learning capacity of the LSTM network).

As described in Section III-A1, the choice of r directly determines the magnitude of the monitoring statistic T^2 and the resultant control limit CL_{T^2} to detect change points, and indirectly dictates the magnitude of the Q statistic and its control limit CL_Q through the remaining residual variates. In existing literature, the optimal r is typically determined based on performance of the downstream learning task [30], [34]. For example, Ruiz-Cárcel *et al.* [34] select r for their fault detection task based on the false alarm rate. For our work, as we leverage CVA in a non-traditional manner to detect device-level change points and enhance RUL estimation, we select r based on the RUL estimation performance.

We assess the $RMSE$ and SF for the RUL estimation produced from the change point integrated model for a reasonable range of candidate $r \in [10, 25]$, as shown in Fig. 9. The value of r yielding the best RUL estimation is selected as the optimal r . Generally, we observe that there is no clear-cut relationship between the r need for good RUL estimation performance and the presence of data complexities such as multiple fault modes or operating conditions. For instance, for both FD001 and FD003 (which differ only in terms of the number of fault modes present), the optimal value of r for achieving the best RUL estimation performance was found to be 15. Similarly, the optimal r for FD002 (which has a single fault mode but multiple operating conditions) was also 15. However, for the most complex dataset FD004, containing both interactions from multiple fault modes and operating conditions, the optimal r achieving the best RUL estimation was larger at $r = 21$. A likely reason for this lack of clear-cut relationship is because temporal variations caused by different fault modes or operating conditions can manifest in either system space captured by r or the “noisier” residual space, depending on characteristic of the fault, the operating condition, and the potential interactions between them. However, a key advantageous aspect of our model is that we do not need in-depth domain knowledge of the fault characteristics or the operating conditions to account for them in the change point detection. As we monitor for breaches in control limits of both the T^2 and Q statistics and conservatively utilise the earlier change point of the two, we can account for significant changes in temporal variations in both the system space and the residual space, regardless of the fault type or operating condition.

Next, we examine the impact of the number of LSTM layers on the RUL estimation performance. Generally, increasing the number of LSTM layers (i.e., depth of the network) increases the model's capacity to learn complex hierarchical relation-

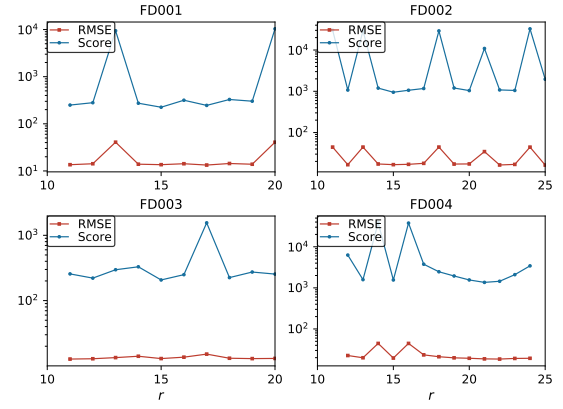


Fig. 9. Impact of number of system canonical variates, r on $RMSE$ and SF .

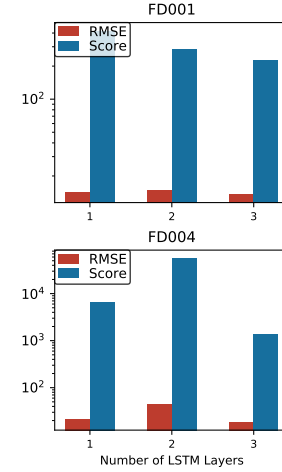


Fig. 10. Impact of number of LSTM layers on $RMSE$ and SF .

ships and dependencies of the data. However, excessive model complexity should be avoided to mitigate the risk of model overfitting. Generally, 2 to 3 LSTM layers are recommended for the C-MAPSS dataset [21], [26], [36]. We assessed the RUL estimation performance for a candidate number of layers in $\{1, 2, 3\}$, and found that the 3-layer configuration yielded the lowest $RMSE$ and SF , as shown in Fig. 10. For the sake of brevity, only the simplest dataset, FD001 and the most complex dataset, FD004 are plotted, but the conclusion on the 3-layer configuration applies to all four datasets.

C. Complexity Analysis

The two key components of our model are CVA-based change point detection and LSTM-based RUL estimation. As seen from Eq. (4), the workhorse of CVA is singular value decomposition (SVD). For the SVD computation, the flop counts (an indication of algorithm speed) increases by $\mathcal{O}(N \cdot p + p^3)$ [47], [48], where N and p are, respectively, the number of observations and number of time lags as defined earlier. The storage space complexity grows by $\mathcal{O}(N + p^2)$ [47], [48]. Next, we discuss the computational complexity of the vanilla LSTM network used in our model. As the LSTM algorithm is local in time and space (i.e., the output of an LSTM cell depends only on the previous cell's output and the

current input), the complexity per weight and time step for updating model weights during training is $\mathcal{O}(1)$ [20]. Given w number of weights, the complexity is thus $\mathcal{O}(w)$.

In comparison, the benchmark models that employ fixed literature values for the piecewise RUL construction can save on the computational complexity associated with a CVA-based change point detection. However, they need to compensate for this “one-size-fits-all” approach with advanced variants of LSTM and hybrid architectures to reach competitive RUL estimation results. These variants also introduce an additional layer of computational complexity to the vanilla LSTM network. For example, the additional complexity of a convolutional layer in GA-CNN-LSTM [45] is $\mathcal{O}(s \cdot L \cdot d^2)$ [49], where s , L , and d are, respectively, the kernel size, sequence length, and the feature representation dimension. Meanwhile, in MC-LSTM [11], the additional complexity of the attention layer [49] is $\mathcal{O}(L^2 \cdot d)$ [49].

Overall, it is evident that all models considered have to incur additional computational complexity beyond vanilla LSTM networks to learn effectively from complex datasets. However, in order to realize the significant practical benefits of change point integrated RUL estimation, we are mindful about restricting the complexity of CVA. For instance, as detailed in Section III-A1, we focus on analysing local temporal dynamics between a limited number of past and future time lags. Thus, $p \ll N$ in the complexity formulation. There are also a growing number of algorithms (e.g., [50]) aimed at reducing the complexity of CVA, which could serve as a basis for the future iterations of our proposed model.

V. CONCLUSION

This paper argues that health status evaluation and change point detection are critical steps for boosting existing RUL estimation model capabilities. In our temporal learning model, we introduce a novel leveraging of canonical variate analysis for degradation monitoring and detecting device-level change points even under varying operating conditions. The proposed method of combining change point detection with LSTM-based RUL estimation outperforms existing models that do not consider heterogeneous change points, especially for Score Function values. Although turbofan engines are used as a case study, the proposed method can be easily generalised to other applications, and be combined with other deep learning RUL estimation models as it is data-driven and does not rely on domain knowledge. Future research will be directed towards extending our unsupervised change point detection methodology to account for shorter lifespan devices with less training data, and further investigating the transition period observed before the degradation state to refine the change points detected.

REFERENCES

- [1] X. S. Si, W. Wang, C. H. Hu, and D. H. Zhou, “Remaining useful life estimation—A review on the statistical data driven approaches,” *European Journal of Operational Research*, vol. 213, no. 1, pp. 1–14, 2011.
- [2] C. Ordóñez, F. S. Lasheras, J. Roca-Pardinas, and F. J. de Cos Juez, “A hybrid ARIMA–SVM model for the study of the remaining useful life of aircraft engines,” *Journal of Computational and Applied Mathematics*, vol. 346, pp. 184–191, 2019.
- [3] M. Wang and J. Wang, “CHMM for tool condition monitoring and remaining useful life prediction,” *The International Journal of Advanced Manufacturing Technology*, vol. 59, no. 5, pp. 463–471, 2012.
- [4] C. Zheng, W. Liu, B. Chen, D. Gao, Y. Cheng, Y. Yang, X. Zhang, S. Li, Z. Huang, and J. Peng, “A data-driven approach for remaining useful life prediction of aircraft engines,” in *21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 184–189.
- [5] Y. Chen, D. Zhang, and W.-a. Zhang, “MSWR-LRCN: A new deep learning approach to remaining useful life estimation of bearings,” *Control Engineering Practice*, vol. 118, p. 104969, 2022.
- [6] J. Zhu, N. Chen, and W. Peng, “Estimation of bearing remaining useful life based on multiscale convolutional neural network,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3208–3216, 2018.
- [7] M. Xia, T. Li, T. Shu, J. Wan, C. W. De Silva, and Z. Wang, “A two-stage approach for the remaining useful life prediction of bearings using deep neural networks,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3703–3711, 2018.
- [8] M. Ma and Z. Mao, “Deep-convolution-based LSTM network for remaining useful life prediction,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1658–1667, 2020.
- [9] S. Wu, Y. Jiang, H. Luo, and S. Yin, “Remaining useful life prediction for ion etching machine cooling system using deep recurrent neural network-based approaches,” *Control Engineering Practice*, vol. 109, p. 104748, 2021.
- [10] W. Huang, H. Khorasgani, C. Gupta, A. Farahat, and S. Zheng, “Remaining useful life estimation for systems with abrupt failures,” in *Annual conference of the PHM society*. September, 2018, pp. 24–27.
- [11] S. Xiang, Y. Qin, J. Luo, H. Pu, and B. Tang, “Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction,” *Reliability Engineering & System Safety*, vol. 216, p. 107927, 2021.
- [12] X. Zhang, Y. Qin, C. Yuen, L. Jayasinghe, and X. Liu, “Time-series regeneration with convolutional recurrent generative adversarial network for remaining useful life estimation,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 6820–6831, 2021.
- [13] L. Liu, X. Song, and Z. Zhou, “Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture,” *Reliability Engineering & System Safety*, vol. 221, p. 108330, 2022.
- [14] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, “Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2306–2318, 2016.
- [15] Y. Mo, L. Li, B. Huang, and X. Li, “Few-shot RUL estimation based on model-agnostic meta-learning,” *Journal of Intelligent Manufacturing*, vol. 34, no. 5, pp. 2359–2372, 2023.
- [16] A. Arunan, Y. Qin, X. Li, and C. Yuen, “A federated learning-based industrial health prognostics for heterogeneous edge devices using matched feature extraction,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–15, 2023.
- [17] K. Liu, Z. Wei, C. Zhang, Y. Shang, R. Teodorescu, and Q.-L. Han, “Towards long lifetime battery: AI-based manufacturing and management,” *IEEE/CAA Journal of Automatica Sinica*, 2022.
- [18] Y. Qin, S. Adams, and C. Yuen, “A transfer learning-based state of charge estimation for lithium-ion battery at varying ambient temperatures,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7304–7315, 2021.
- [19] Z. Chen, L. Chen, W. Shen, and K. Xu, “Remaining useful life prediction of lithium-ion battery via a sequence decomposition and deep learning integrated approach,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1466–1479, 2021.
- [20] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [21] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, “Long short-term memory network for remaining useful life estimation,” in *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2017, pp. 88–95.
- [22] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, and X. Li, “Machine remaining useful life prediction via an attention-based deep learning approach,” *IEEE Transactions on Industrial Electronics*, vol. 68, no. 3, pp. 2521–2531, 2020.

- [23] C. G. Huang, H. Z. Huang, and Y. F. Li, "A bidirectional LSTM prognostics method under multiple operational conditions," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8792–8802, 2019.
- [24] Y. Liao, L. Zhang, and C. Liu, "Uncertainty prediction of remaining useful life using long short-term memory network based on bootstrap method," in *2018 International Conference on Prognostics and Health Management*. IEEE, 2018, pp. 1–8.
- [25] Z. Wu, S. Yu, X. Zhu, Y. Ji, and M. Pecht, "A weighted deep domain adaptation method for industrial fault prognostics according to prior distribution of complex working conditions," *IEEE Access*, vol. 7, pp. 139 802–139 814, 2019.
- [26] Z. Shi and A. Chehade, "A dual-LSTM framework combining change point detection and remaining useful life prediction," *Reliability Engineering & System Safety*, vol. 205, p. 107257, 2021.
- [27] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *2008 international conference on prognostics and health management*. IEEE, 2008, pp. 1–6.
- [28] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, 2018.
- [29] S. Greenbank and D. A. Howey, "Piecewise-linear modelling with automated feature selection for li-ion battery end-of-life prognosis," *Mechanical Systems and Signal Processing*, vol. 184, p. 109612, 2023.
- [30] P. E. P. Odiowei and Y. Cao, "Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 1, pp. 36–45, 2009.
- [31] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 International Conference on Prognostics and Health Management*. IEEE, 2008, pp. 1–9.
- [32] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.
- [33] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [34] C. Ruiz-Cárcel, Y. Cao, D. Mba, L. Lao, and R. Samuel, "Statistical process monitoring of a multiphase flow facility," *Control Engineering Practice*, vol. 42, pp. 74–88, 2015.
- [35] S. J. Kim, S. H. Kim, H. M. Lee, S. H. Lim, G.-Y. Kwon, and Y.-J. Shin, "State of health estimation of li-ion batteries using multi-input lstm with optimal sequence length," in *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*. IEEE, 2020, pp. 1336–1341.
- [36] J. Zhang, P. Wang, R. Yan, and R. X. Gao, "Deep learning for improved system remaining life prediction," *Procedia Cirp*, vol. 72, pp. 1033–1038, 2018.
- [37] T. Tieleman and G. Hinton, "Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning," *COURSERA Neural Networks Mach. Learn*, vol. 17, 2012.
- [38] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.
- [39] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade: Second Edition*. Springer, 2012, pp. 437–478.
- [40] P. Baldi and P. J. Sadowski, "Understanding dropout," *Advances in neural information processing systems*, vol. 26, 2013.
- [41] F. Li, L. Zhang, B. Chen, D. Gao, Y. Cheng, X. Zhang, Y. Yang, K. Gao, Z. Huang, and J. Peng, "A light gradient boosting machine for remaining useful life estimation of aircraft engines," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3562–3567.
- [42] Z. Ma, J. Guo, S. Mao, and T. Gu, "An interpretability research of the XGBoost algorithm in remaining useful life prediction," in *2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*. IEEE, 2020, pp. 433–438.
- [43] C. Zhao, X. Huang, Y. Li, and S. Li, "A novel cap-LSTM model for remaining useful life prediction," *IEEE Sensors Journal*, vol. 21, no. 20, pp. 23 498–23 509, 2021.
- [44] A. Boujamza and S. L. Elhaq, "Attention-based LSTM for remaining useful life estimation of aircraft engines," *IFAC-PapersOnLine*, vol. 55, no. 12, pp. 450–455, 2022.
- [45] U. Amin and K. D. Kumar, "Remaining useful life prediction of aircraft engines using hybrid model based on artificial intelligence techniques," in *2021 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2021, pp. 1–10.
- [46] M. Sayah, D. Guebli, Z. Nouredine, and Z. Al Masry, "Deep LSTM enhancement for RUL prediction using Gaussian mixture models," *Automatic Control and Computer Sciences*, vol. 55, pp. 15–25, 2021.
- [47] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2000.
- [48] W. E. Larimore, "Canonical variate analysis in control and signal processing," *Statistical methods in control and signal processing*, pp. 83–120, 1997.
- [49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [50] X. Fu, K. Huang, E. E. Papalexakis, H.-A. Song, P. P. Talukdar, N. D. Sidiropoulos, C. Faloutsos, and T. Mitchell, "Efficient and distributed algorithms for large-scale generalized canonical correlations analysis," in *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 2016, pp. 871–876.

APPENDIX A

COMPARISON OF CURRENT WORK AGAINST EXISTING LITERATURE ON CHANGE POINT-INFORMED RUL ESTIMATION

See table IV.

APPENDIX B

INVESTIGATION ON MINIMUM LIFESPAN OF TRAIN ENGINES

As discussed in Section IV-A1, the choice of the minimum lifespan needed for train engines of the change point detection model influences the CLs learnt, change points detected, and ultimately the RUL estimation. We consider a range of candidate minimum lifespans {100, 125, 150, 175, 200, 225} to assess the appropriate minimum lifespan needed for train engines to produce well-performing RUL estimations (which is a strong indicator for the quality of change points learnt).

For the sake of brevity, the analysis focuses on the "worst-case" scenarios of FD001 and FD003, which have only 100 train engines to begin with. For FD001, the lifespans of the train engines range from 128 to 362 operation cycles, with an average lifespan of 206 cycles. For FD003, the lifespans range from 145 to 525 cycles, with an average lifespan of 247 cycles. Table V presents the *RMSE* and *SF* values of the RUL estimations produced from the various candidate values for the minimum lifespan needed for the train engines. We observe that the RUL estimation performance generally improves as the minimum lifespan requirements for the train engines increases. At the minimum lifespan threshold of 200 cycles, there is a significant improvement in the RUL estimation performance. This confirms that the validity of our initial assumption on the first 60 operational cycles being from normal operation is indeed stronger for train engines with at least 200 operational cycles. Interestingly, increasing the minimum lifespan requirement beyond 200 cycles worsens the RUL estimation as it considerably reduces the number of suitable train engines available for training the change point detection model.

TABLE IV
COMPARISON OF OUR WORK WITH EXISTING LITERATURE ON CHANGE POINT-INFORMED RUL ESTIMATION.

	Shi and Chehade [26]	Wu <i>et al.</i> [28]	Ours	Remarks
Are the standard C-MAPSS test datasets used for evaluation?	x	x	✓	Existing works do not use the standard predefined test datasets, which are more challenging as the sensor data is available up to only some abrupt time before failure and the lifespan information is unknown. Instead, existing works use a portion of the train engines (with the full lifespan information known) for testing.
Is the RUL estimation performed independent of the equipment's lifecycle stage?	x	x	✓	Existing works limit their RUL estimation and evaluation to only the last 50 cycles before failure. However, we perform RUL estimation on all test engines of various lifecycle stages and do not restrict our evaluation to only late-stage RUL estimation.
Is the RUL estimation evaluated under multiple operating conditions?	x	✓	✓	Shi and Chehade [26] evaluate their method on the FD001 and FD003 datasets, which only have a single operating condition.
Size of test data used for evaluation	Small (10 engines)	Small (20 engines)	Large	Our work utilises the full, standard, pre-defined test engine dataset.
Are standard performance evaluation metrics used?	✓	x	✓	Instead of the RMSE metric, Wu <i>et al.</i> [28] report the relative prediction error, which refer to percentages of samples in the testing set with relative prediction errors less than or equal to 5%, 10%, and 20% respectively.
Is the analysis of change points detected interpretable?	x	x	✓	Our change point detection method is highly interpretable as the monitoring statistics and departures from control limits can be easily visualised and monitored.

TABLE V
IMPACT OF MINIMUM ENGINE LIFESPAN CHOSEN ON DOWNSTREAM RUL ESTIMATION (BEST IN **BOLD**).

Min. train engine lifespan	FD001		FD003	
	<i>RMSE</i>	<i>SF</i>	<i>RMSE</i>	<i>SF</i>
100	NA ¹	NA ¹	17.77	1197.28
125	NA ¹	NA ¹	17.77	1197.28
150	NA ¹	NA ¹	16.18	408.92
175	15.89	304.10	15.43	502.75
200	13.59	224.88	12.94	207.10
225	40.69	11749.48	40.18	13031.71

¹ NA indicates that the chosen minimum lifespan was too short to detect change points.