Domain-driven Residual Useful Life Estimation

Navid Zaman[1], Daniel Chan[1] and Chris Stecki[1]

[1] *PHM Technology, 120 Queens Parade, North Fitzroy, Victoria, 3068, Melbourne, Australia*

Abstract

Predictive maintenance is widely agreed upon as the superior method for maintenance scheduling over time/condition-based choices. This is where the remaining life cycles of each part/component as well as the system in its entirety are forecast, informing the optimal period to perform a maintenance action. In current literature, this is often done using a value or set of values representing the degradation of the components or system; such a value is called a Health Index (HI), built from a function of the sensors on the machine(s) to be monitored. This paper will introduce and expand upon methods to utilise a flexible and domain-defined (HI), derived from the use of a Digital Risk Twin (DRT). The twin is used to capture domain knowledge directly from subject matter experts to determine the optimal HI at both the component stages in addition to one's representative of the system, thus defining at various levels of indenture. The HI is coupled with state-of-the-art deep learning and machine learning techniques to confidently forecast trends observed in a system or its individual parts to enable prognostics with a high calibre of predictive integrity. Syndrome Diagnostics (SD), a tool to incorporate the research and prototypical work laid out will also be presented.

Keywords: predictive maintenance, health index, residual useful life, digital risk twin, machine learning, deep learning, syndrome diagnostics

Introduction

An important part of Predictive Maintenance (PdM) is the ability to reliably and accurately forecast the remaining life of a component or system to build maintenance schedules that are efficient. To this effect, the emerging technology of machine learning (ML) and deep learning (DL) have been widely adopted. However, quick embrace of these methods involves risk in form of spurious correlation and the hefty demand of information required from the domain at various parts of the process, especially at the prediction result level, where ideally minimal interpretation should be required.

Using a bearing system, this paper will demonstrate the use of timely collection of domain data which informs the rest of the process and how this information is utilized in improving the quality of ML/DL algorithm predictions. The approach focuses on understanding the problem component/system and then using state-of-the-art technology to solve its remaining life estimation. Much of the methodology in this paper follows the book [1] closely.

Body of the Paper

Engineering Context

Residual Useful Life is defined as the amount of time the component within a system can operate for before it reaches a failure that requires attention in the form of maintenance actions. Maintenance is often scheduled to refresh the life of the component; however, it is a form of preventative maintenance and is sub-optimal as there may be significant life remaining for the maintained component. Maintenance performed on the component once the failure has occurred is often detrimental to the operations as it often costs the most.

To ensure the system is available at any given time, it is important to calculate the residual useful life to accurately schedule an optimized predictive maintenance program.

- Predictive vs Planned vs Reactive
- Cost of downtime
- Maintenance and cost optimization

Capacity and Duty

The remaining life of any machine is dependent on the area between its (load-carrying) capacity and duty (load) curves [3]. A failure occurs when these curves intersect, and as such, the prediction and forecast of these variables are integral to the estimation of the remaining useful life.

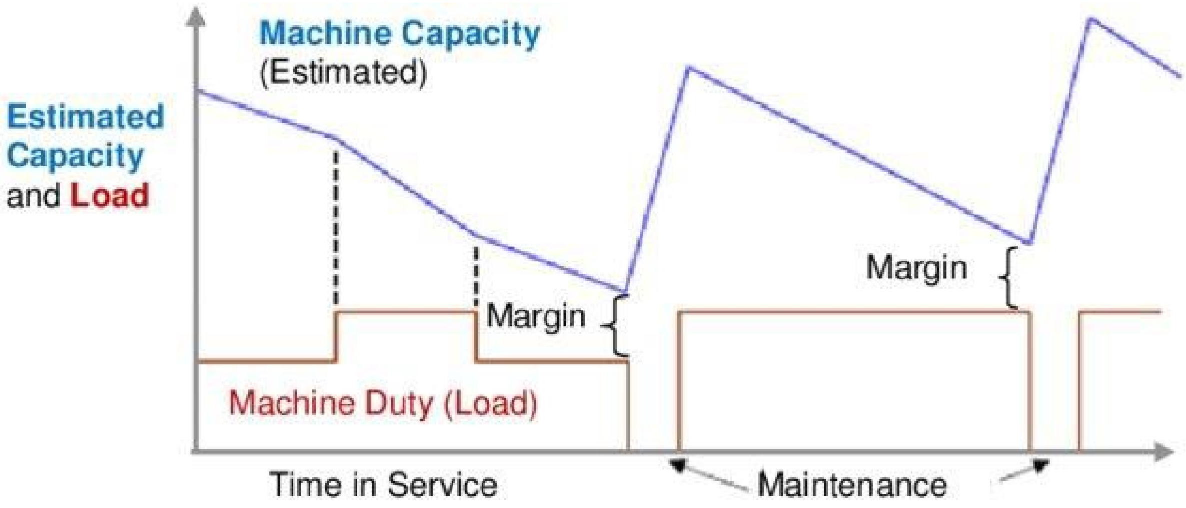

Fig. 1: Scheduled maintenance capacity-duty curves

In this paper, the focus is mainly on the estimation of the Health Index, which abstracts the duty curve. Capacity estimation is a function of age and loading.

The Digital Risk Twin: Definition of the Digital Risk Twin (DRT)

The definition of a digital twin will vary between organizations. Many digital twins exist which are used for different purposes. In the context of engineering, the digital risk twin (DRT) is defined as a computerized representation of a system which enables engineering analyses, such as fault trees, reliability assessments and Failure Mode and Effects Analysis (FMEA), to be performed to assess the potential exposure to harm or failure. This DRT will capture the functional behaviour and dependencies of the system as well as the physical failures using standardized taxonomy to ensure data consistency and mitigate ambiguities.

The Digital Risk Twin: Sensor Placements and PHM

As engineering moves towards predictive maintenance, it is essential for the systems to have some form of diagnostic capability to assess the health-state of the system. Typically, this involves implementing sensors to monitor specific parameters of components which are then post processed and correlated to historical data to identify potential failures. In most cases, the failure detection can only be assessed once a failure has occurred where the data is available to be analysed.

Sensor placements are determined in the initial stages of product development with the aim to collect data for the diagnostic capability. Typically, these sensor placements are determined based on historical data and laboratory tests of failed components. Though these sensors placed on systems may be sufficient, they are often isolated to detect a single failure and are not optimized for system diagnostics.

The Digital Risk Twin: Degradation states

For a healthy component, the HI value usually remains stationary. However, as soon as the component develops a fault, the HI value begins to increase or decrease. Depending upon the rate of defect propagation and degradation growth, the HI can exhibit different trends. To cope with the problem of RUL prediction, it is crucial to capture the degradation trend from time-series data. As shown in fig.1 we divide the degradation trend into two stages: the healthy stage and the accelerated degradation stage. The prediction of RUL is triggered as soon as a bearing exhibits signs of degradation and starts departing from its steady-state behaviour. The instance time of when the prediction process is initiated is term as the Incipient Point.

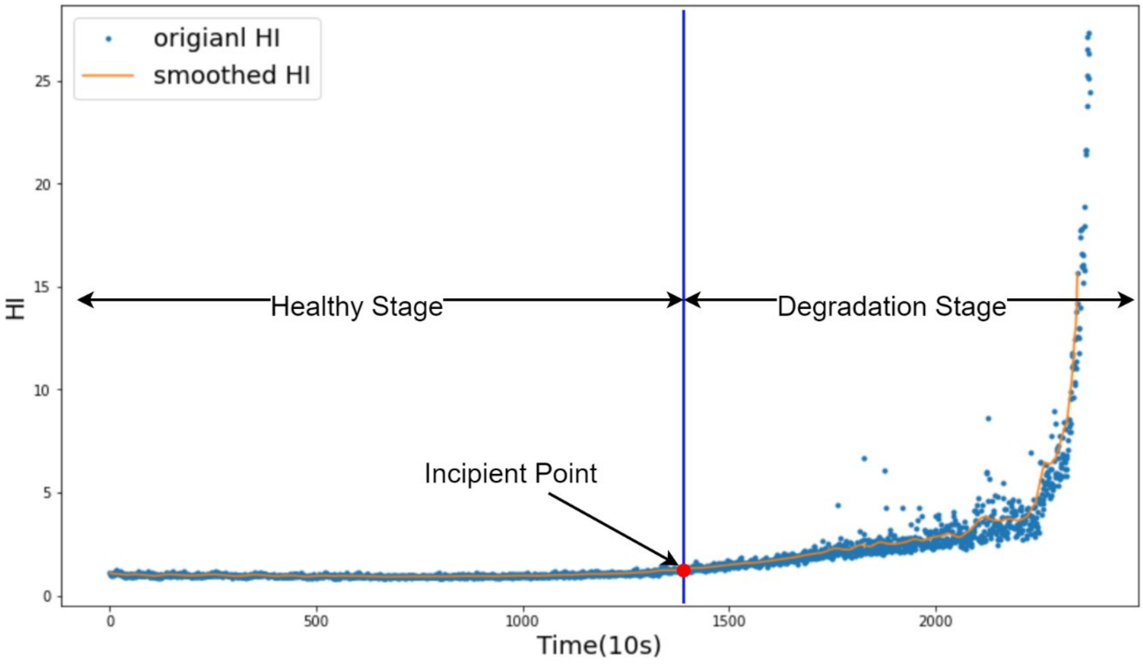

Fig. 2: Common HI trajectory for a component

HI (Health Index) and it's usage: Definition of Health Index

The Health Index is an expression or function of the sensors on the component to determine the degradation or health of the component. It is a domain informed value that is captured from a subject matter expert (SME) ahead of time. This reduces some of the guesswork or interpretation requirements from engineers later in the process, after correlated predictions are made.

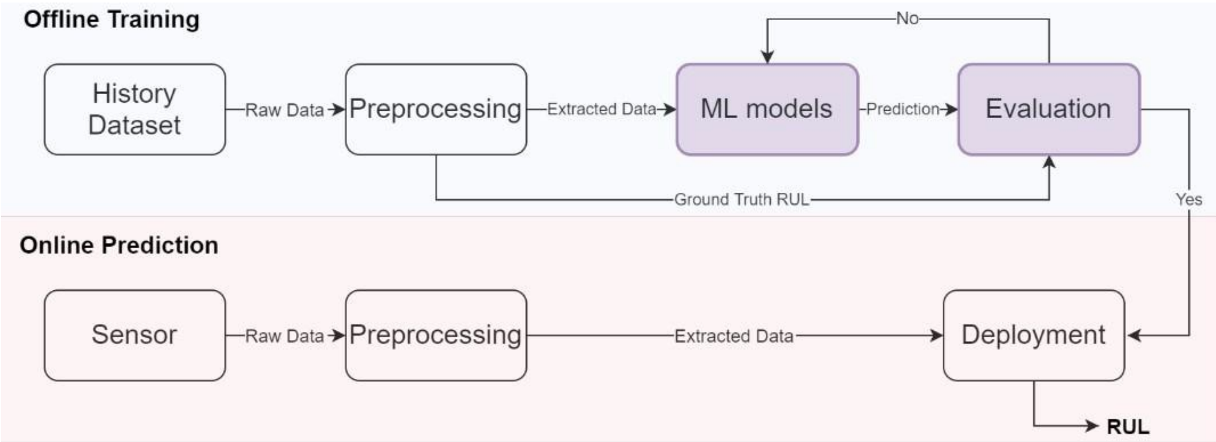

Fig. 3.1: Traditional RUL preparation online and offline

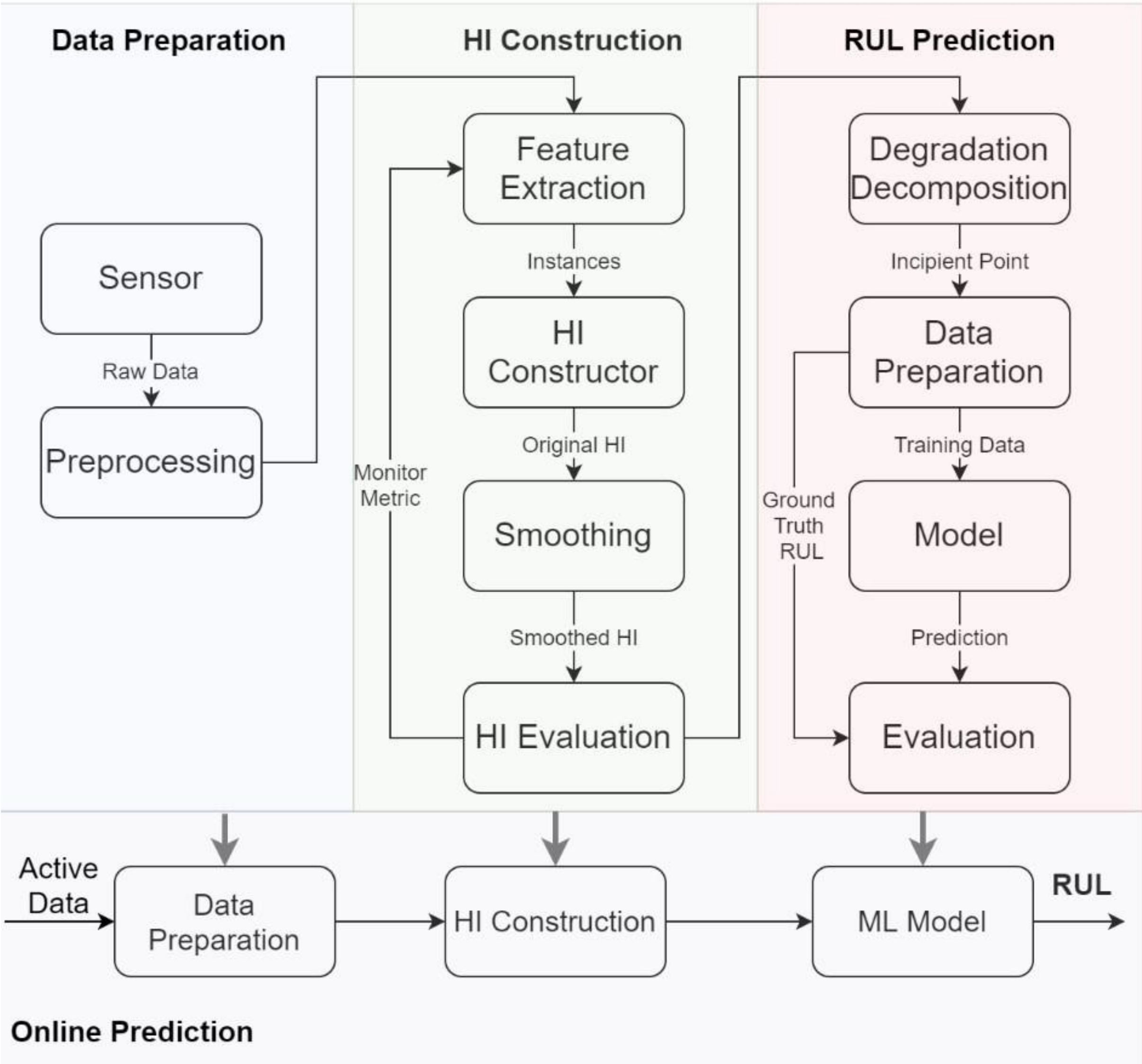

Fig. 3.2: Overview of the method planned to use in this paper

HI (Health Index) and it's usage: Classical and HI-based RUL

The classical RUL prediction process is the general regression-based solution to predict the RUL by supervised machine learning model. The advantage of this model is that it needs a little domain knowledge to build the model. We use random forest regressor, support vector regressor and long short-term memory recurrent neural network to predict the remaining useful life of the components. However, HI-based RUL prediction process is another regression-based solution to predict the RUL. The HI is introduced to infer the component's health condition and it also brings domain knowledge to the prediction model. Hence, the HI- based RUL prediction process is more flexible and easier to implement. In the experiment, we explore the HI construction and RUL prediction.

How the Index is calculated: Feature Extraction

Raw sensors often contain more than is required for the calculation of HI. Such disturbances such as noise, inconsistent sampling rates and poor scaling throw off any prediction that would need to be made. In addition, much of the data remaining after a cleaning process contributes little to the quality of analysis. All the issues can be tackled via extraction of useful features from the data.

Feature extraction, which generates new features from the pre-processed time-series data, aims to reduce the massive amount of data into a manageable synoptic data structure and preserve the character of the original data as much as possible. When the input data is too large to be processed and it is suspected to be redundant, it can be transformed into a reduced set of features. In the RUL estimation model, the extracted feature should reflect the consistency of the performance degradation curve of the component. Diverse types of feature extraction methods will be implemented for diverse types of original time-series data. Therefore, a broad way of feature extraction methods should be provided. In our bearing dataset experiment, three families of feature extraction techniques: time domain, frequency domain and time-frequency domain are used to extract meaningful features from signals. A list of the extraction methods are in table 1.

Table 1: Comparison of feature extraction methods

Name	Domain	Metric
WPT standard deviation of inverse hyperbolic sine at 0 ~ 1/16 fs	Wavelet Packet	0.9310
WPT 5 percentile data at 0 ~ 1/16 fs	Wavelet Packet	0.9297
WPT energy level at 0 ~ 1/16 fs	Wavelet Packet	0.9266
fd_{p10}	Frequency Domain	0.9237
WPT 75 percentile data at 0 ~ 1/16 fs	Wavelet Packet	0.9215
WPT energy ratios at 3/16~ 1/4 fs	Wavelet Packet	0.8911
WPT energy ratios at 7/16 ~ 1/2 fs	Wavelet Packet	0.8887
WPT energy ratios at 3/8 ~ 7/16 fs	Wavelet Packet	0.8830
fd_{p6}	Frequency Domain	0.8818

How the Index is calculated: Construction of HI

These indices are derived from the domain understanding of subject experts and as such, there needs to be a process for a user to define what the index is. This is done via an expression of the available sensors and various operations. In fig.3, we demonstrate several ways to describe HI's where each feature is a sensor value. The operators used to build the expression can range from simple addition and multiplication to extraction functions mentioned prior.

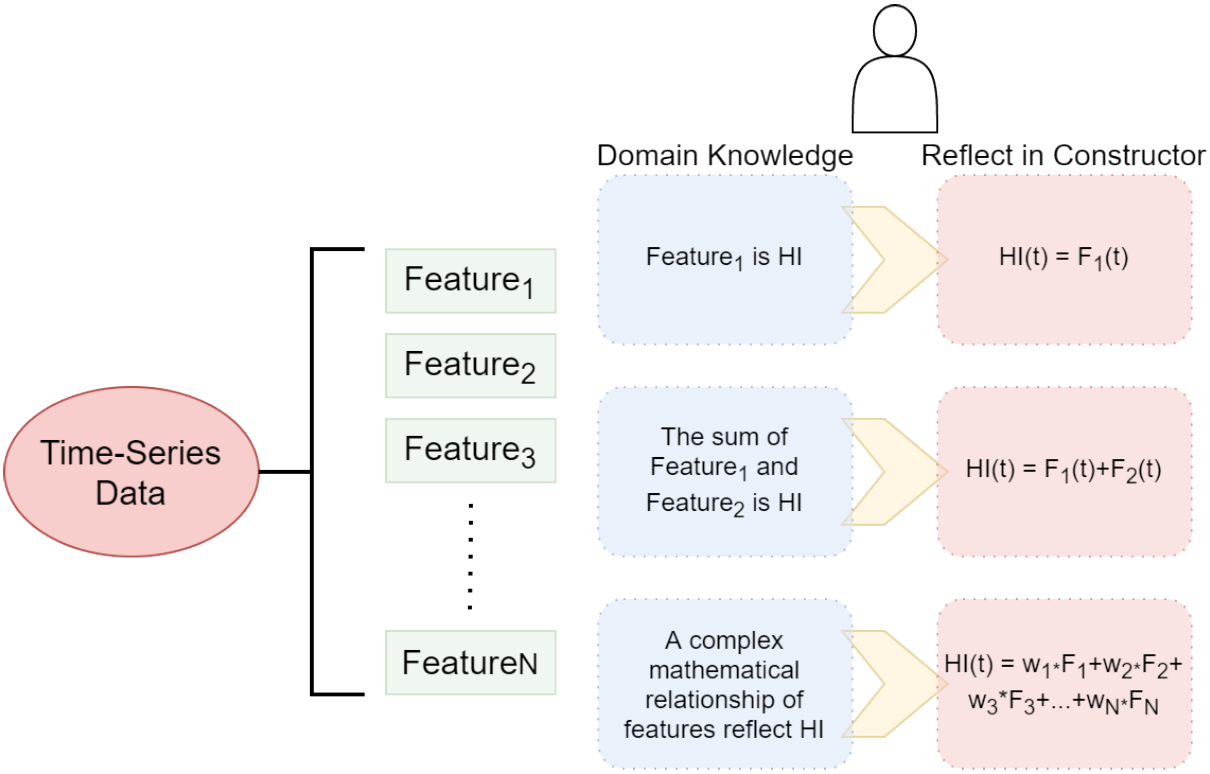

Fig. 4: Possible ways to construct a HI

How the Index is calculated: Smoothing

Normally, the value of HI undergoes frequent fluctuation (noise) despite exhibiting a certain underlying trend over time. These fluctuations (noise) require smoothing to reveal the underlying trend and to construct a reliable model that can be used to make a reasonable RUL estimation. Furthermore, a well-designed smoothing method can increase the effectiveness of HI (for example, convert HI into a monotonically increasing function). There are several common smoothing techniques: exponential smoothing techniques, moving average and weighted smoother etc. In this experiment, we use a so-called Linear rectification technique (LRT) cascade by a moving average smoother. The smoothing of the health indicator is illustrated in the equations 1 and 2

$$y_i \begin{cases} y_i & \forall y_{\{i-1\}} \leq y_i \; (1 + \eta) y_{\{i-1\}} \\ y_{\{i-1\}} + 1 - \eta & \forall y \; < y_{\{i-1\}} > (1 + \eta) y_{\{i-1\}} \end{cases} \quad (1)$$

$$\eta = \frac{1}{n} \left| \sum_{\{i=1\}}^{n} y_{\{i+1\}} - y_{\{i\}} \right|$$

(2)

How the Index is calculated: Evaluation of HI

Even though the indices may be obtained from a domain expert, it does not necessarily mean that the quality of predictions will be correct. There may be more than one solution of HI that exists; an ideal index expects to have the ability to track the degradation trends over time across sufficient fault signatures. Three evaluation metrics are used to measure effectiveness. These metrics are manually designed by considering the physical characteristic of HI.

- The correlation metric (trendability) measures a linear correlation between HI and operating time, and it is calculated by equation 3.

$$Corr = \frac{\sum_{\{i=1\}}^{n} (H_{\{i\}} - \bar{H})(t_{\{i\}} - \bar{t})}{\sqrt{\sum_{\{i=1\}}^{n} (H_{\{i\}} - \bar{H})^2 \sum_{\{i=1\}}^{n} (t_{\{i\}} - \bar{t})^2}}$$

(3)

where H_i and t_i are the health indicator and time values of the i-th observation sample. n is the length of the samples during the lifetime.

- The monotonicity metric evaluates an increasing or decreasing trend of the HI. It bases on the assumption that the health status of a component would not recover if there is no repair or other interventions. The monotonicity is measured by the absolute difference of positive and negative derivatives as shown in equation 4.

$$Mon = mean\left(\left|\frac{\{\#\Delta_{\text{pos}}\}}{\{n-1\}} - \frac{\#\Delta_{\{negative\}}}{\{n-1\}}\right|\right)$$

(4)

- The robustness reflects the degree of fluctuation of the HI. It indicates the tolerance of a HI to outliers, and it is calculated by equation 5.

$$Rob = \frac{1}{n} \sum_{\{i=1\}}^{n} \exp\left(-\left|\frac{HR_{\{i\}}}{H_i}\right|\right)$$

(5)

where $\mathrm{HR} = \mathrm{H} - \grave{}\mathrm{H}$ and $\grave{}H$ is obtained by the causal moving average filter.

- It can be noted that all the three assessment indexes are confined in the range of [0,1]. To assess the overall ability of HI, a composite index (CI) that contains all the three indexes is defined as in equation 6.

$$CI = 0.95 * Corr + 0.05 * Mon + 0.05 * Rob$$

(6)

Algorithms and methods in forecasting of HI: Data

The data used for the experiments in this paper are from the bearing dataset presented for the PHME IEEE 2012 Data Challenge [2]. This dataset consists of 3 bearings, each containing a temperature and vibration sensor. This has been organized into 17 sets, each run to failure and correct failure times recorded.

Algorithms and methods in forecasting of HI: Algorithms

There are two types of ML model approaches:

1. The first approach is the direct prediction for time series forecasting. For every single HI value, there is a corresponding RUL target value. The learning algorithm finds patterns in the training data that maps the input HI to the target RUL, and it outputs an ML model that captures these patterns. In the experiment, support vector regressor (SVR) and XGboost are used.
2. The second approach is the sliding window approach for time series forecasting. Since time series forecasting problems have a sequential nature or can be processed sequentially, it is easy to consider using a bunch of HI to predict the RUL. In other words, we use more than a one-time variable to predict the RUL. This method reflects the assumption that previous health conditions may also impact current health status. A simple LTSM with a window size 60 is used.

Algorithms and methods in forecasting of HI: Metrics

The performance is evaluated by two metrics. The first metric is the root mean square error (RMSE). The second metric is a scoring function. The percentage error on experiment i is defined by equation 7.

$$Er_i = ActualRUL_i - \frac{PredictRUL_i}{ActualRUL_i} \tag{7}$$

With deduction to early RUL prediction and more severe deduction for RUL estimates that exceeded actual component RUL. The score of the accuracy of RUL for experiment i is defined by equation 8.

$$score \begin{cases} \exp^{-\ln(0.5)*(\frac{ER_i}{5})} & Er_i \leq 0 \\ \exp^{+\ln(0.5)*(\frac{Er_i}{20})} & Er_i > 0 \end{cases} \tag{8}$$

The final score of RUL prediction has been defined as the average of all experiment's scores as defined in equation 9.

$$score = \frac{1}{n} \sum_{\{i=1\}}^{n} score_i \tag{9}$$

The continuous prediction from the incipient point (the component begins to accelerate degradation) to the end of running life for three ML models are shown in the below figures. As we can observe, the SVR and XGboost predicted RUL shows abnormal results - this is because of the lack of training data. Since we use the sigmoid activation function in LTSM, it does not show this problem.

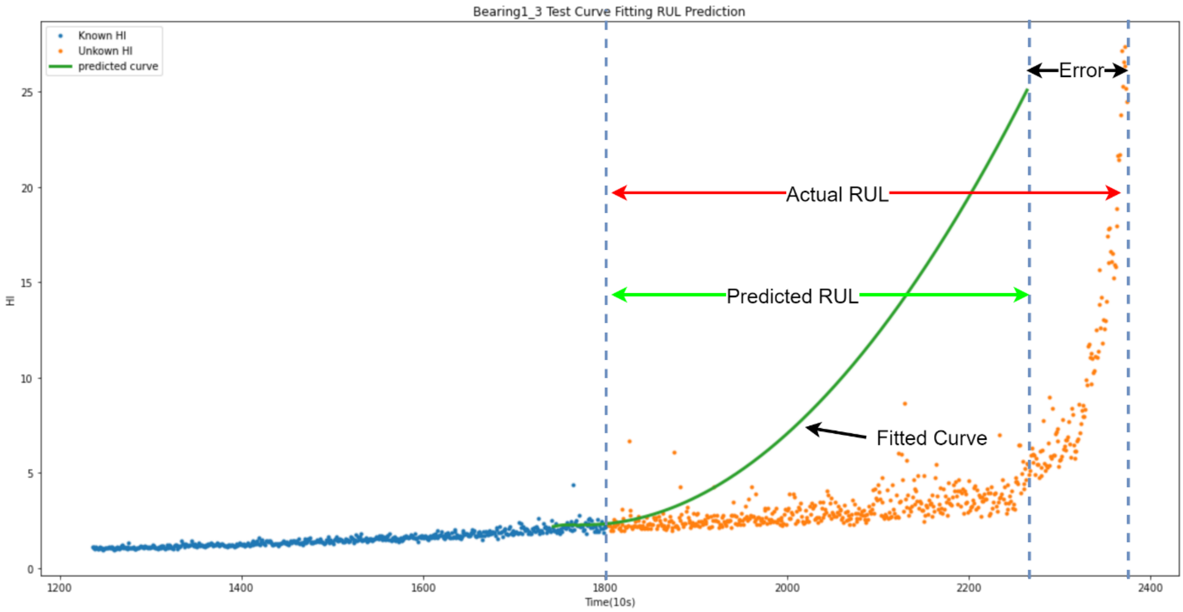

Fig. 5: Example of the fitted curve from examples

Table 2: Metrics for the incipient point

Method	**Sliding Window used?**	**RMSE**	**Score**
SVR	NO	0.0398	0.97
XGBoost	NO	0.0304	0.95
LSTM	YES	0.0355	0.99

Table 3: Metrics for bearing dataset

Method	**Predicted**	**Actual**	**Error**	**Score**
Curve fitting	492	573	14.14	0.995
SVR	481		16.06	0.994
XGBoost	593		-3.49	0.995
LSTM	498		13.09	0.995

Conclusion

Predicting residual useful life is a crucial step in prognostics and in PdM overall, enabling smarter maintenance schedules, reducing both downtime and costs of failure. Most strategies in this vein of work utilize powerful algorithms to correlate between signals to divine the remaining life of the sensed component/system. This paper has introduced the usage of a risk twin to inject domain knowledge to the prediction to improve reliability and reduce the need for interpretation from the user. Using the DRT, we also automate aspects such as having available a library of HI's (features specific to a component/subsystem that tells us the degradation) and ability to determine a system-wide RUL via an aggregation. In the future, this is expected to be added to our diagnostics tool, Syndrome Diagnostics as a part of a live analysis and reporting ecosystem.

References

1. Kim, N., An, D., & Choi, J. (Eds.). (2017). Prognostics and health management of engineering systems. Springer.
2. Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., & Varnier, C. (n.d.). (Tech. Rep.). IEEE, Title = An experimental platform for bearings accelerated degradation tests. In IEEE International Conference on Prognostics and Health Management, Year = 2012
3. Akinyemi, Olasunkanmi. (2010). Development of Equipment Maintenance Strategy for Critical Equipment. Pacific Journal of Science and Technology. 11. 328-342.