Treatment of non-monotonic trends in fault-progression of turbo-fan engine

MSc Thesis

Sumant Malekar

Treatment of non-monotonic trends in fault-progression of turbo-fan engine

MSc Thesis

by

Sumant Malekar

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday, 16th of May 2022 at 10 am

Student number:	4374762
Project duration:	December 2020 – May 2022
Thesis Committee:	
Chair	Dr. B.F. Santos
Supervisor	Dr. M.L. Baptista
Examiners	Dr. D. Zarouchas
	Dr. A. Bombelli

TU Delft

Acknowledgements

The number of aircrafts in the sky are only going to rise in the coming future. Concerns related to their safety and costs are increasing with it. This research addresses these concerns related to aircraft maintenance and contributes to the advance data-driven methods with experimentation to improve the accuracy of prognostics. This document concludes my work for my master's degree in Aerospace Engineering and my journey at the Delft University of Technology. The process of thesis has expanded my knowledge of data-driven prognostics and made me curious to learn more about the topic.

I would like to thank the people who supported me through the process. This research would not have been possible without the support and guidance of my supervisor, Dr.ir.Marcia Baptista. I would like to take this opportunity to thank her for giving me the responsibility of my own research and helping me to find the right path and motivation through the research. I am also very grateful to Dr.ir.Bruno Santos, not just for the constructive feedback and informed suggestions on project but also for being understanding and supportive throughout my thesis. I will cherish every discussion and learnings from both. I am also grateful to all the remarkable professors that have shaped me throughout my Bachelor's and Master's degree.

Thank you to all my family. My parents, Anantkumar and Sujata Malekar, and my brother Shounak, for their relentless encouragement, care, support and patients with my thesis. I would particularly thank my parents for providing me with such an opportunity and devoting their attention to help me with my goals. I would also like to thank my girlfriend Akshaya for her support and keeping me aligned with my thesis always. Lastly, I would like to thank my former roommates, my thesis friends, and all other friends from Delft and Enschede. This is all thanks to you.

Sumant Malekar
Delft, May 2022

Contents

Introduction

Regular maintenance of an aircraft assists the airline to monitor its health while maintaining the operations of the airline overall. However, with unplanned maintenance and component failures, the operations are disturbed and have a negative impact on the airline's expenses. To identify these failures beforehand, predictive methods are used to solve such issues. Physics-based methods have been effective, but they are expensive and require immense experience. With the growing data-driven techniques for maintaining the health of a system, the maintenance activities are cost effective and do not require in-depth knowledge of each component. The predictions are made based on a collective data processed from identical components and a trained model to predict the behavior of component's life. The models are usually used to predict the remaining useful life (RUL) of the components to plan maintenance before its failure/damage. With this method, it is possible for aircraft maintenance to carry out its operations with safety and cost effectiveness.

Although, the data-based models do not predict RULs accurately. This causes a state of worry for implementing such methods as having wrong predictions would result in heavy damage to the airlines and a concern for safety. Extensive research and experimentation are carried out to better the accuracy of predictions and continue to improve the technology. The goal is to predict RUL on time, so it is not dangerous and still cost effective. This project aims to further add value to this goal. Fault progression data can have unwanted and unnecessary information, which is not required for a machine learning (ML) algorithm to learn. This extra data can be treated with monotonic constraints and then cleaner data can be injected into the ML model. The main objective of the project is:

To check if posing monotonicity on the data before inputting in the ML model would improve the predictions of RUL for fault progression

The above objectives will be addressed in this report with the following structure. The Part I presents a scientific paper that amalgamates the literature review required for the topic, the experimentation carried out, the methodology followed, and evaluation of results. Part II deep dives in the literature review and provides background information on prognostics and case studies. Lastly, Part III gives an overview of the initially planned methodology to address the goals and objective of the research.

I

Scientific Paper

Treatment of non-monotonic trends in fault progression of turbofan engine

Sumant Malekar
Chair: dr. ir. B.F. Santos, Supervisor: dr. M. Baptista
Delft University of Technology, Delft, The Netherlands

This research aims to investigate and experiment on a state-of-the-art problem to treat the non-monotonic behavior of fault progression trends in predictive maintenance. Well-established algorithms and literature are researched for fault progression prognostics, however, not considerable attention has been given to monotonic constraints at a preprocessing stage. A non-monotonic trend carries complex information which has outliers and nonessential signal values. The goal of the project is to motivate the usage of monotonic constraints to treat non-monotonic signals of a degrading component. The problem is presented as follows: Determining if the monotonic constrained method at a preprocessing step shall assist prognostics to estimate the remaining useful life of a component accurately. To explore this research, a monotonic constraint - Average Conditional Displacement (ACD) is used at the preprocessing step of a model, in comparison with regular preprocessing methods. The model is experimented on the NASAs simulated C-MAPSS datasets of turbofan engine and modelled with two prognostics algorithms. The model performance is measured with performance metrics. The results showcase that by treating non-monotonic trends with monotonic constraints does improve the prognostics. However, they are not significantly advanced compared to other preprocessing steps.

I. Introduction

Scheduling incorrect maintenance of flights has costed substantial damage to the airline industry [1]. Disasters have occurred because of an equipment failure or a component of an aircraft which was not monitored properly. It is vital to predict the occurring failures of components and structures before time. Maintenance activities have to further consider regular operation of flight, crew availability, availability of labor for maintenance, and ensuring business stability; the scheduling of maintenance activities needs to be precise. Forecasting capability and strategically planning are the reason for the precision of scheduling. With the growing involvement of data-driven methods in aircraft maintenance, a significant number of researchers are developing methods to improve prognostics.

Predictive maintenance has evolved from a standalone niche case to fast-growing, high return on investment applications delivering the right value. The data volume has risen significantly. Its market size is predicted to grow to 28 billion in the next 5 years [2]. It shows how important and cost-effective data-driven techniques have become for predictive maintenance, along with other methods (like physics-of-failure-based). Data-driven approaches are more suitable and do not rely on any domain knowledge of a component. However, these methods are not yet in power with the physics of failure yet. There are many advanced technologies developed in data-driven methods for airline maintenance everyday, yet it is critical to optimize the prediction further in the next few years to meet the demand of growing operations [3].

This paper discusses an important step for improving the prognostic of fault progression. A component's life degrades after every use and ultimately reaches to the end of life (EoL). However, signals extracted from the sensors of the component are not entirely insightful for prediction [4]. The data is scattered and has no clear patterns that could be used for ML models to learn form. Even though the health of the components would only degrade over time, uncertainties and inefficient sensors could disturb the montone behavior of signals and induce an unwanted spike in the health of the component. The ML model would learn and imbed this behavior and eventually predict incorrect RULs. Therefore, to avoid such discrepancies and non-monotonic behavior, the author explores extraction of the monotonic trend from the fault progression before the model is trained with an ML algorithm. The hypothesis is that having a monotonically constrained signals would improve the training of ML models and eventually generate better prognostics. The objective of the paper is to identify and establish the benefits of constraining a non-monotonic fault progression signals to a monotonic signals to avoid uncertainty and improve predictions.

The remainder of the paper follows the given structure. Section II gives an overview of the literature and related work on monotonicity of signals. Section III describes the datasets and evaluation metrics used for the experiment. Section IV explains the steps taken for preprocessing and modeling of the algorithms. Section V displays the results and analysis. Finally, Section VI summarizes the paper and concludes with recommendations for future work.

II. Background

The end goal of prognostics, specifically degradation-based algorithms, is to accurately predict the RUL of individual components based on their performance and use. Degradation measurements could be sensed from temperature or vibration level, or inferred measurements like model residuals and physics-based model predictions [5]. These performances can be compared to the trends of extracted features from raw data to serve the requirements of degradation modeling. In a real world scenario, the performance of the prognostic model are due to the critical issues limiting its applicability. For example, its acceptable uncertainty level, human interventions, and expected accuracy [6]. These aspects must be considered for prognostics models.

The data of degrading component consists of signals and stochastic variations (noise). The noise is a feature which has to be minimized during the feature extraction [7]. Feature selection requires a large amount of exhaustive research for choosing the relevant features [8]. There are well-researched feature extraction tools (like [7][8]) to display clear degradation trends. However, comparatively less research and attention has been given to non-monotonic degradation features in prefailure repairs. These repairs could be a result of even human intervention. Nonlinear behaviors do not indicate the state of the machinery under operations [6]. Unplanned maintenance activities can also change the usual degradation signals and hence cause non-monotonic fault progression of signals. Developing uncertainty in signals lead to the ML algorithm deviating from predicting accurately [9].

A. Non-monotonic & Monotonic signals

Non-monotonic signals are noisy and there is no consistent change in the mean level of degradation [10]. The trends of a non-monotonic signal carry complex information which is not entirely insightful for prediction. The data are scattered and do not necessarily follow a pattern for a ML model to train and replicate for a fault progression process.

Uncertainty in signals has been an issue for successful PHM models [9]. According to Baraldi [11], imperfections in the predictability of prognostic models can be caused from three different sources. Randomness related to future degradation, modeling errors and inaccuracies in degradation data. This paper focuses on inaccuracies in degradation signals, primarily due to non-monotonic fault progression and compares it to the effect of constraining on monotonic fault progression. Research papers like [8] express how monotonicity in data can avoid the exhaustive search without sacrificing optimality. It is also recommended by [5] and [12] to quantify and consider monotonicity in the systematic construction of PHM models. Feelders [13] mentions that models trained on monotonic datasets often have better predictive performance than models trained on original data. Monotonic datasets could be created by generating artificial data or by relabeling of real data [14][15][16].

A monotonic signal is defined by the unidirectional and consistent change in the mean level of degradation data. Monotonicity as a property states that an increase in input cannot result in a decrease in the related output [10]. So, adding a monotonic constraint over a model would reconstruct the signals to guarantee a monotone relation between explanatory variables and dependent variables. Monotonicity poses a monotonic trend where the condition is fulfilled if and only if the trend is either entirely non-increasing or entirely non-decreasing. In other words, if the signal is monotonically increasing or decreasing with time, corresponding to an improving or deteriorating system, there is supposed to be a monotone trend, otherwise the trend is non-monotone [17]. This behavior of the trend would be useful for prognostics as the signals would reduce the uncertainty and deduce the actual life cycle of the component for the algorithm to learn and predict (as can be seen in Fig. 1). It was also shown by [18] and [19] in their experiments that monotone prediction outperforms the standard counterparts due to successfully avoiding overfitting.

On the contrary, Ben-David [20] mentions that adding monotonic constraints to ordinal regressors can reduce accuracy. To counter the hypothesis, the author of [10] says that the explanatory variables in the datasets used for the experiments by Ben-david were not in a monotone rela-

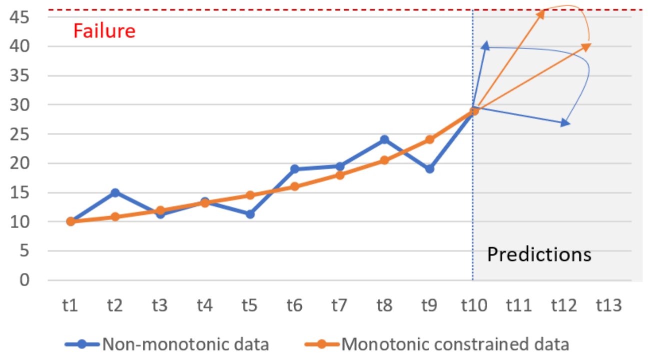

Figure 1. Monotonic vs non-monotonic signals and the hypothesis of prediction windows of RUL values

tionship with the labels. One can take away from this discussion that making use of monotone models inappropriately can result in a poor performance. The author [10] further says that the benefits of monotone or partially monotone data models can be fully beneficial when one can be sure of the relations present in the data.

B. Monotonic improvement techniques

There are techniques available to improve the monotonicity of data and they are discussed in the following. Monotonic classification and regression model approaches have been proposed in the specialized literature [20] [21] [22]. Also, Monotonic neural networks [23][24] and hybridizations. The authors of [18] show a class of neural networks that is monotone. This is obtained with nonnegative weights and a multilayer neural network.

Ben-david [25], proposes an attribute selection metric which considers the monotonicity and error while building decision trees. The paper shows how the metric reduces the non-monotonicity of decision trees while maintaining the inductive accuracy. Ben-david also discusses monotonic classification models in the paper [20]. He discusses how adding monotonicity can help algorithms learn and impair their accuracy.

As ranking functions are used for decision making applications, the authors of [26] propose transparent participation metrics to clarify the ranking process of monotonicity. There are authors who have tried to improve the monotonicity of data in the data mining process. The author of [27] describes the measures to express the degree of monotonicity of data and an alogirthm to clean non-monotonicity. The algorithm relabels the dependent variable in non-monotonic datasets and transforms into a monotonic dataset. The paper also shows the best methods to enforce monotonicity of decision models. In the paper [28], the author Barile also focuses on data mining algorithms which enforce montonic restrictions while learning from data.

These models were successful in predicting accurate prognostics compared to their counterparts with non-monotonic models. The question arises if one has the datasets constrained on monotonicity already before injecting into a ML-model, how different would the performance be compared to the dataset that was not constrained? In theory, the simplification and filteration beforehand would reduce the computational time and yield the results of predictions in simple steps. Having a clean signal against time and its simple monotonic behavior, would be easier to learn than an uncertain one. However, there is possibility of missing out valuable information and could reduce the overall prediction.

C. Monotonic constraints in preprocessing

Preprocessing step in prognostics has been important for reducing randomness in data for injecting the right information to the algorithm to reduce discripencies. This process reduces the computational time and increases accuracy in prognostics. There are different methods of preprocessing. For example, feature engineering is a process where unclassified and unlabeled data are clustered into a group with similar trends or examples. It results in identifying correlations and dependencies existing between their features [29]. Similarly, for analyzing features, Principal component analysis (PCA) is available. This method's goal is to analyze the covariance and reduce the complexity of data. In [9], the PCA was used for smoothing acoustic emission signals by eliminating highly correlated variables. Smoothing is an extracting technique applied to time series to extract variations between time steps [29].

Preprocessing step is also responsible to reduce the noise in the signal trends. To identify and learn the behavior of a signal during the training of the model, the high noise in the signal would be inco-operated and studied. However, for degrading material or identifying the RUL of a material, the noise factor does not add a value to identify the EoL. It increases the computational steps and the window of prediction. There are methods available which help to reduce the noise, such as data cleansing, rolling mean or moving average.

Rolling mean operations results in cleaner and understandable trends for the ML model to learn and perform prognostics accurately. However, these processed trends still have considerable distortion and noise [29]. Moving average develops a new series with average values of raw data along the time series. While computing, it assumes the time series to be stationary and the signal does not have monotonic behavior or seasonality. The moving average depends on the window width defined by the raw observations and calculates the average per window width [29].

Regarding the monotonic behavior of data, monotonicity poses a trend where the condition is fulfilled if and only if the trend is either entirely nonincreasing or entirely nondecreasing. The series is unidirectional too. Implementing monotonicity to the signals will result in the updated signal. As the component's state replaces a healthy to an unhealthy state, the features of the component show a degrading trend in the data (with time). In other words, changing monotonously in the decreasing propagation. The monotonic constraint used in this thesis is discussed in the following.

The algorithm average conditional displacement (ACD), was introduced by Vamos [30]. The algorithm is based on a signal value interval and it is able to estimate the monotonic trends of a stationary noise-filled time series data [9]. The algorithm has two folding advantages. First, this algorithm does not require any initial subjective assumptions and approximates monotonic trends as a piecewise linear curve by dividing into subintervals of signal intervals [30]. Secondly, it is an automated algorithm which is comparable with known methods like moving average and polynomial fitting [9]. The numerical quantities of ACD is presented in the following literature [30][31].

The algorithm considers small intervals randomly placed across a signal. The extreme ends of the intervals are selected and an average line is drawn through the points. This is illustrated in the Fig. 2. The figure describes a small interval of the pieces of x_n that enter into the computation of sample average. Whereas the thick continuous red-line denotes the ACD approximation of monotonic trend from an interval of $(X_a, X_{a+1}]$. The sample average slope of the given points in the interval can be estimated by the Eq. 1. Where $\hat{g}$ is the slope and N_j is the number of x_n values of interval $(X_a, X_{a+1}]$. Increased number of N results in improved accuracy of trend estimation. The iterations of intervals are carried out

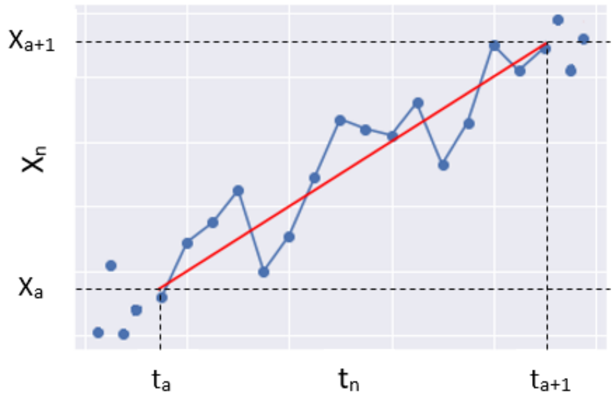

Figure 2. A step variation of time series where the thick line describes the ACD approximation of monotonic trend [9]

throughout the trend. Resulting in a smooth monotonic curve. The author [9] says the ratio between the noise fluctuation and the amplitudes of trend variation is the major contributor to the accuracy of the ACD algorithm. ACD has the potential benefits to treat non-monotonicity and stationary noise in PHM applications.

$$\hat{g} = \frac{1}{2N_a}\left(\sum_{x_n \epsilon I_a} \delta x_n + \sum_{x_{n+1} \epsilon I_a} \delta x_n\right) \tag{1}$$

Imposing a monotonic constraint on the non-monotonic signal and improving the monotonicity of the dataset would be discussed in the following. The question, "Can a monotonic constraint impose monotonicity on a nonmonotonic fault regression and improve prognostic accuracy eventually?" is primerily important for the research, It will be answered in the Section IV. "Can a montonic constraint as a preprocessing method add value to the prognostics compared to other methods?" is another important question to be answered.

III. Experimental framework

A. C-MAPSS Dataset

The dataset chosen for the experiment is the NASA developed simulated Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset [32]. The dataset was developed for prognostics research with actual RUL values available to use, hence run-to-failure data sets and fault evolution failure. The dataset comprises of simulated engine degradation sampling multivariate sensor data over a given time cycle, divided into 4 different sub-datasets. Each dataset has different operational conditions and fault modes. Training, Testing, and RUL sets are available for each sub-dataset, each containing a set number of engines. The operation of the engine deteriorates with increasing cycles until the end of life (EOL). This information is captured by 21 available sensors on the engine's components.

For training sets, all available sensor values from the engines can be used for training and testing samples. While for testing sets, only one data point corresponds to the last recorded operational cycle. Hence, it is only used for test samples. In summary, the training sample depends on both the number of engines and the number of data points per engine, while the number of test samples only depends on the number of engines [32]. The dataset used is train-FD001 for the research. The research is limited to one operating condition and fault mode. FD001's application fits with the purpose of experiments. The goal of the topic does not require to perform on each dataset and hence it would have been time consuming to use all.

B. Monotonic metric

Coble [5], describes Local Gradient-Based (LGB) method as, "The average difference of the fraction of positive and negative derivatives of the time series of a feature over time". The formula used to compute the monotonicity metric is given by Eq. 2. Where G^+ and G^- represent vector positive and negative derivatives respectively (seen in Eq. 3). y_i is the value of the sensor at a given time and N represents the total number of sensor points in the interval.

$$\mathrm{MM}_1 = \mathrm{mean}\left(\left|\frac{\#\left[G^+\right]}{N-1} - \frac{\#\left[G^-\right]}{N-1}\right|\right) \tag{2}$$

where,

$$\begin{aligned} G_i^+ &= \frac{\Delta y_i^+}{\Delta t_i} = \frac{y_i - y_{i-1}}{\Delta t} \mid (y_i - y_{i-1}) > 0 \\ G_i^- &= \frac{\Delta y_i^-}{\Delta t_i} = \frac{y_i - y_{i-1}}{\Delta t} \mid (y_i - y_{i-1}) < 0 \end{aligned} \tag{3}$$

The same formula can be rewritten as Eq. 4[5]. The differences of adjacent points of a signal are compared for identifying monotonicity. N_j is representing the total points, $x_j(k)$ represents a point of the signal and $x_j(k+1)$ is the subsequent point. Each step of the signal is compared and an average score is determined. For each trend, the monotonicity adjusts from a value of zero to one. Zero being the least monotonic and one with the highest monotonicity. The formula would be used for checking the monotonicity of the signal. It will also be useful for comparing different preprocessing datasets for their monotonicity check.

$$\text{monotonicity} = \frac{1}{M}\sum_{j=1}^{M}\left|\sum_{k=1}^{N_j-1}\frac{\mathrm{sgn}\left(x_j(k+1) - x_j(k)\right)}{N_j - 1}\right| \tag{4}$$

C. Performance metric

For evaluating predictions from the model, the following performance metrics would be useful for analysing and proving hypotheses. Firstly, the error while predicting RUL is calculated by subtracting the true RUL from the predicted RUL value. The errors are bound to be either positive and negative. Having a positive error (late prediction) would be more damaging as the component could fail before the predicted time, hence the scoring function penalizes the positive error [33]. The scoring function is given by Eq. 5, where h_i represents the prediction error and N the test sample size. The scoring function is further developed to estimate the total score over a complete dataset. The score value is multiplied with the number of test units and divided by testing samples. This score value is useful while estimating the last sample for comparing the complete set [32].

$$S = \begin{cases} \sum_{i=1}^{N}\left(e^{-\frac{h_i}{13}} - 1\right) & \text{for } h_i < 0 \\ \sum_{i=1}^{N}\left(e^{\frac{h_i}{10}} - 1\right) & \text{for } h_i \geq 0 \end{cases} \tag{5}$$

Mean Absolute Error (MAE) is the average magnitude of the errors of prediction. The error scales linearly and reports for each diversion of prediction are made. MAE is defined by Eq. 6[34].

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|h_i - \hat{h}_i| \tag{6}$$

Root Mean Square Error (RMSE) is given by the Eq. 7. It penalizes early and late prediction errors equally and indications how many time cycles the predictions are off on average [32]. Furthermore, the error is squared and hence the weight of the error would be larger compared to MAE.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} h_j^2} \tag{7}$$

Coefficient of determination or the regression score function (R^2) is an asymmetric function. It indicates what proportion of our dependent variable can be explained by the independent variables we use. The best value for R2 would be 1 and the least value can be negative.

To evaluate the accuracy of the models and compare the overall accuracy with varying bounds over time, the alpha-lambda accuracy is selected. The quality of a prediction falls within specific limits at particular times with respect to a performance measure [35]. The bounds of the cones are determined and the score is calculated based on the Eq. 8, where α is accuracy modifier and λ is the time modifier.

$$\alpha - \lambda Accuracy = \begin{cases} 1, (1+\alpha) \cdot RUL \leq \text{Predicted } RUL \geq (1-\alpha) \cdot RUL \\ 0, \text{ otherwise} \end{cases} \tag{8}$$

IV. Methodology

The steps taken for the research problem are discussed in this section. The data of the case study discussed in the section III will be used for the experiment. Two ML-algorithms and 3 preprocessing methods are compared in the experiment.

A. Preprocessing

To understand the effectiveness of proposed preprocessing method, the models are carried out parallely with four different methods. Applying a rolling mean to the raw data (original data), applying exponential moving average (EMV) to the raw data, monotonic constraints (ACD in this case) on the raw data and the raw data itself. The preprocessing operators discussed in the Section II, are implemented individually during this step. However, the first step involves selecting the available sensors for applying filters. The sensor 1, 5, 6, 10, 16, 18, and 19 in dataset FD001 exhibit a constant value throughout the life of the engine. Firstly, a constant value does not add value to the algorithm [31]. Secondly, as ACD filter is unable to produce monotonic trends for seasonal trends, these sensors are excluded from the filtering dataset. After processing into a monotonic behavior, the sensors are added back to the dataset.

Similarly, the original FD001 training dataset is filtered with rolling mean and exponential moving average. An engine's time cycle for each dataset is compared with their monotonic metric. For these four data types, following were the results shown in Table 1. The ACD filter applied to the raw dataset shows the optimal result for a monotonic constraint, while the raw dataset, rolling mean and exponential moving average dataset have a significantly smaller monotonic value. The results can be visualised in Fig. 3, which represents a sensor value trend for the different types of datasets. The table and the figure confirm that ACD filter managed to monotonically constraint the signals. The trend in the figure also shows the upward heading curve throughout. Hence, it manages to comply with the rules of monotonicity and essentially answers the first part of question addressed in Section II.

Table 1. Coble's monotonic metric [5] of sensor 2 (T21) unit 1 of training datasets

Dataset	Monotonic metric
Raw data	0.068
EMV	0.089
Rolling mean	0.111
ACD	1

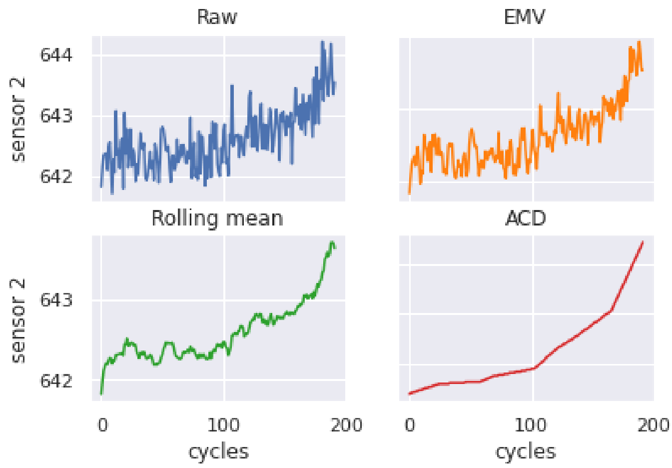

Figure 3. Sensor plots of each data type, sensor 2 (T21) engine 1

The rolling mean and exponential moving average data types have lesser noise compared to the original signal (Raw data). Although, the rolling mean is much closer to the monotonic constraint signal than the exponential moving average signal. Furthermore, these modified data types and the original dataset would be compared in the following steps for their predictions and performances. These labels will be used throughout the report to maintain consistency.

B. Splitting train-test dataset

Splitting of dataset is used to understand the performance of ML-models and evaluate the behavior of the results. The available training dataset is split into training and testing dataset. The test dataset can be used for following the trends of the predicted value and each sensor value can be compared. It also helps to identify the RUL error at each signal value for this experiment. The FD001 training dataset will be split into into 80-20% training and testing datasets, respectively. For the size of the dataset and the overall variance can be presented for testing within the 20% of the dataset. The preprocessing methods will be applied to the training datasets and the testing dataset (raw dataset) will be common to all methods for analysing the output.

C. ML-algorithm

Two algorithms are chosen for experimentation. First, the random forest regression model is chosen as it should be a good fit for providing accurate results for the large time-series dataset [36]. It is also a common and widely used regression model. Secondly, a combination model of convolution neural network (CNN) and Long-short-term memory (LSTM) model is implemented based on the work of [33]. The paper experiments on CMAPSS dataset and shows the benefit of using the best of both types of neural networks for estimating RUL. CNN model exceeds in extracting features and LSTM is capable of building long-term time dependencies. They both are used in combination in the paper of [33]. To avoid unnecessary noise, the paper uses 1D convolution to extract features and feeds it to LSTM for learning long-term dependencies. This neural network model is trained and tested similar to the random forest algorithm. The numerical results and an error visualisation for the model are shown in the section V. This model is also used to validate the results from the first model.

V. Results

This section discusses the results obtained after applying the two models on 4 different dataset types. Performance evaluations are also made in this section.

A. Performance metric for random forest regressor

The numerical results of the RF regressor can be seen in the Table 2. This table represents the performance of the RF regressor for each data type. The error calculated from the difference of predicted RULs over the actual RULs is represented with various metrics. As can be seen from the table, the regressor performs best on the ACD filtered dataset when comparing the mean metric amongst the rest. The R^2 score is higher for the ACD filter too. It is not a large difference but it is noticeable how the error is higher for the datasets with less monotonic value already.

Table 2. Numerical metrics of RF regressor for each data type

Data type	MAE	RMSE	R^2
Raw data	32.52	47.85	0.63
EMV	32.11	47.91	0.63
Rolling mean	32.10	49.59	0.61
ACD	31.28	49.64	0.64

The visualisation in Fig. 4 describes the RUL prediction compared to the actual groundtruths for the test engine number 19. It is observable how close the trends are to the groundtruth. However, the ACD type has signals which are alternatively diverging and converging towards the groundtruth. Which is not a good sign. This example shows how the ACD causes errors in the prediction because of the monotonic data that fed into the model. Although, the second half of the predictions are stable and are closer to the groundtruth. To identify the size of errors in terms of fitting of the predictions compared to groundtruth, the scoring function is used further for analysis.

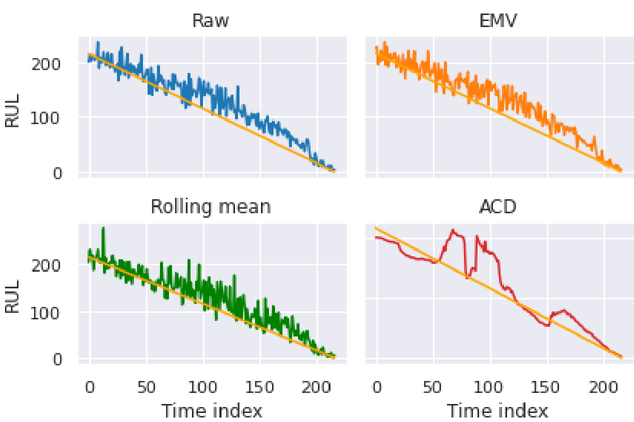

Figure 4. RUL prediction vs time for engine number 19

The error between the predicted and the actual RULs is substituted in the scoring function formula discussed in the section III. The errors with bigger magnitude increase the scoring function. The visualisation for the error can be seen in the Fig. 5. The score for the predictions is seen in the Table 3.

The average score for the engine unit 19 shows that it is highest for the rolling mean and then followed with ACD. Interestingly and as observed in the Fig. 4, the ACD signal doesn't match to the ground truth. However, maintaining the rules of montonicity, it sticks out through its middle phase and maintains distance from the groundtruth.

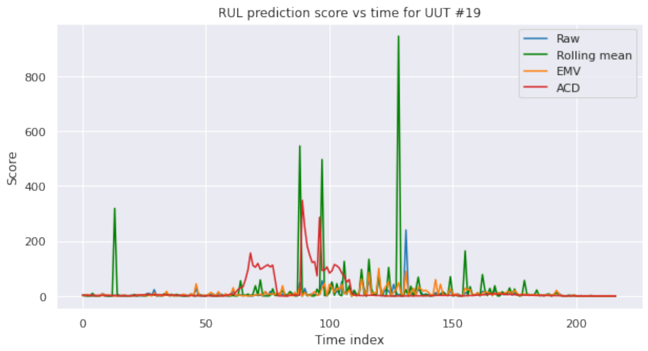

Figure 5. RUL prediction score vs time for engine number 19

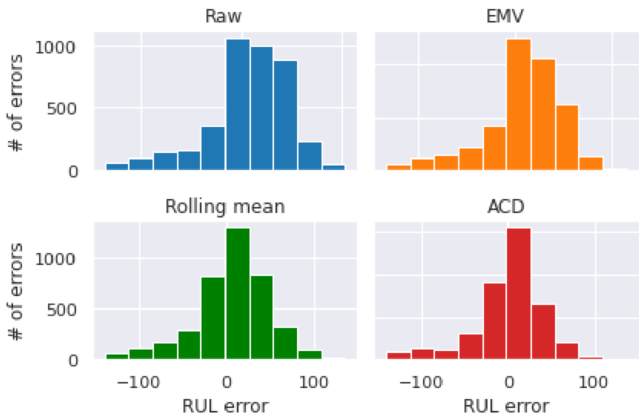

Figure 6. Distribution of number of errors to identify the early/late predictions

Table 3. Scoring function and overall average score

Data type	Average score engine no. 19	Overall average score for 20 test engines
Raw data	8.05	3575.37
EMV	9.811	4386.18
Rolling mean	20.69	2872.70
ACD	20.48	9551.14

Causing larger gap and hence increasing the score value. Furthermore, this can be reflected for all the 20 test units as the overall average score column represented in the Table 3.

The score value is the lowest for the raw dataset, which indicates less amount of noise and non-monotonicity. The reason is the calculation steps of the scoring function. The calculations for underestimates and overestimates add the scoring value depending on which side of the groundtruth the prediction value lies on. If the RUL prediction is higher for underestimation, the values would be negative. To check how the RULs have been predicted, a simple distribution histogram is visualised to check if the predictions are made earlier or later than the actual RULs.

The Fig. 6 is an histogram of prediction errors. The errors occuring to the left of "0 RUL error" (x-axis) are early estimates of RULs and to its right are the late estimates. The early estimates are better than the late estimates as it will help to plan a maintenance activity before failure. However, being closer to the on-time prediction point (0-error) would yield the best results for prognostics. As seen in Fig. 6, the magnitude of ACD datasets are maximum near the on-time prediction. Raw dataset and rolling mean are also close. However, their values of errors are higher on the right of on-time performance, whereas it is considerably low for ACD. This result suggests that ACD filter has performed better in its competition.

The discussed metrics showcase the performance of the model. To understand how accurately the regressor predicts RUL for each dataset type, the alpha-lambda curves are utilized. The graph is presented with the following variables, the alpha '+' bound is represented by the blue straight line, and the alpha '-' is represented by the orange straight line. The dotted green line passing through the middle of alpha '+' and alpha '-' is the groundtruth line. Predicted points lying inside the bounds are counted as accurate. The end tapers as the value of RUL has to be predicted accurately as possible. For the examples used in the figures below, alpha was chosen as 0.3 as it was the standard value used. However, this value was varied after this step for further analysis.

For the same engine test unit 19, the alpha-lambda bounds are superimposed as seen in the Fig. 7 on the RUL prediction vs time. The prediction for this set of unit is lower in terms of metric as seen earlier. With the total points lying inside the bounds, an overall accuracy is calculated for each dataset and is seen in the Table 4. As seen in the visualisation, the ACD has the lowest accuracy. The reason for this inaccuracy is the same as mentioned before. However, an interesting observation can be made towards the end of the time index. The ACD manages to predict very close to the RUL value. Which is the most important requirement for the model. To further analyse and compare, engine test unit 16 is visualised.

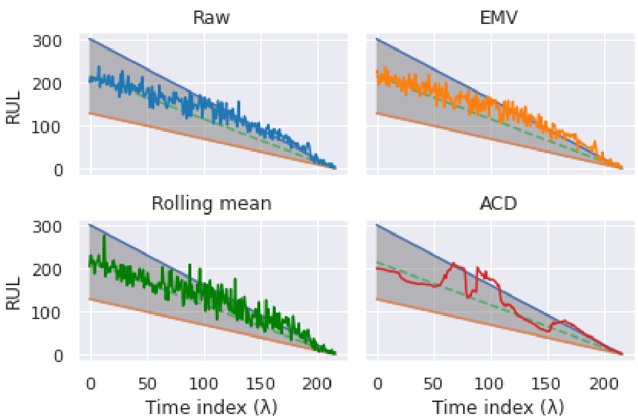

Figure 7. Alpha-lambda metric of engine unit no. 19. $\alpha = 0.3, \lambda = 0.5$

Table 4. Overall accuracy of engine unit no. 19. $\alpha = 0.3$, $\lambda = 0.5$

Data type	Overall accuracy
Raw data	70.96 %
EMV	68.66 %
Rolling mean	73.27 %
ACD	67.74 %

As can be seen from the Fig. 8 and the accuracy values in Table 5, the datatypes do not perform according to the previous metrics. The visualisation in this figure shows how predictions from ACD datasets have steared towards the groundtruth and maintained it throughtout life cycle. The model was able to predict the trend and align the RUL prediction closer to the actual RUL. The table also shows the accuracy of the ACD dataset prediction to be maximum. In cases as such, the real potential of monotonic constrained filters can be noticed. The prediction accuracy increased and the performance eventually would be better for these outputs. The prediction from the raw data did not manage to stay under the alpha-lambda bounds until the quarter of the time index.

Two engine cases with different accuracy results were observed. All test engines need to be examined for their accuracy and an overall result has to be concluded. Therefore, a combination of visualisation for the accuracy of each test engine at each time interval is developed. The visualisation would not only help to understand the overall accuracy for all test units, but it would also benefit in understanding if the accuracy of prediction is high or low

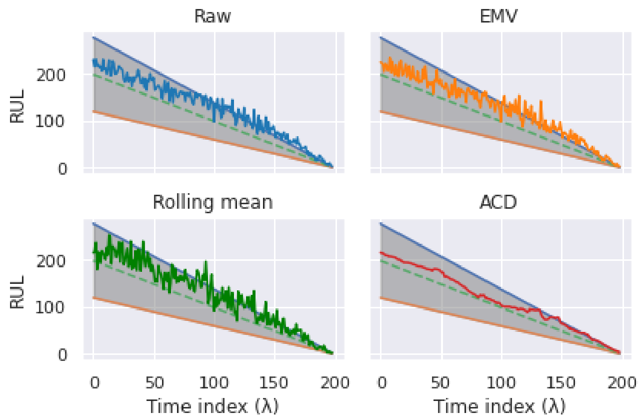

Figure 8. Alpha-lambda metric of engine unit number 16. $\alpha = 0.3$, $\lambda = 0.5$

Table 5. Overall accuracy of engine unit no. 16. $\alpha = 0.3$, $\lambda = 0.5$

Data type	Overall accuracy
Raw data	64.32 %
EMV	63.81 %
Rolling mean	70.35 %
ACD	82.41 %

at a given point in time. The Fig. 9 describes the overall prediction accuracy over time intervals as the percentage of component's life for alpha is 0.3. The bar chart shows the accuracy for each dataset and compares its accuracy for each time interval.

The prediction accuracy of ACD dataset is maximum throughout (mostly). Initially, the accuracy is low and the models are out of bounds. Nevertheless, the overall prediction accuracy of the test samples is highest for ACD dataset. Again, proving monotonic constraint datasets helped the predictions to be more accurate.

To conclude, one last step is taken to measure the accuracy of predictions with changing alpha (from 0.1 to 0.9). Each alpha value is iterated over the test engine and the results are observed in the Fig. 10. Here it is possible to assess the accuracy of the models over cone size and percentage of accuracy. The predictions from ACD dataset outperforms other dataset predictions. Except for the alpha of 0.1, the ACD dataset manages to help the regressor to predict accurate RULs. Which means, with very precise and small bounds, the predictions of ACD dataset are not as accurate. The reason is that the RUL predictions are not as accurate in the first half of the time

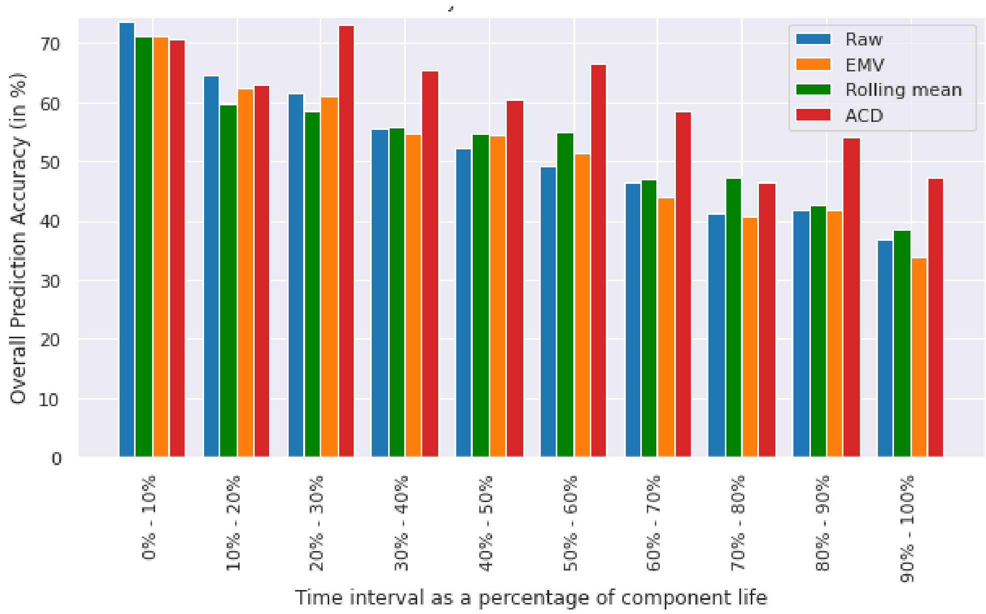

Figure 9. Overall prediction accuracy as a function of time intervals for each datatype. $\alpha = 0.3$ and $\lambda = 0.5$

intervals from the ACD datasets The ACD filter observes the extreme ends of the trends and apply monotonic constraints to restructure the signal. The trained model picks up values which are off-setted from the groundtruth and eventually causes it to underfit the predictions too.

In conclusion, insights in the process of adding a pre-processing monotonic constraint are seen and interesting results and outcomes are derived. Deep performance understanding of the predictions from ACD filtered dataset is made and compared with known preprocessing methods. The comparisons helped to understand how better or worse the ACD performed. For the next step, a complex algorithm is modelled (as mentioned in Section III) with the same dataset to analyse the results and validate the process.

B. Performance metrics of CNN-LSTM model

The CNN-LSTM model depends on the combination of temporal convolution layers and LSTM layers with data augmentation. The target function is followed accurately by the LSTM. This algorithm has a good accuracy and effective prediction. With increasing number of epochs, the weights are changed in a neural network and the curves go from underfitting to optimal curves. Thus, a constant number of 100 epochs were used. This could be a solution to the underfitting predictions of ACD filters from the previous model and improve the prediction accuracy overall. The splitted dataset (mentioned in section IV, similar to the RF model, is loaded into the CNN-LSTM model. The results are discussed in the subsequent section. The test dataset FD001 is also performed on and experimented in Appendix B.

Table 6. Numerical metrics of CNN-LSTM for each data type

Data type	MAE	RMSE	R^2
Raw data	21.02	35.51	0.8
EMV	15.07	23.33	0.89
Rolling mean	14.59	24.97	0.87
ACD	14.18	25.45	0.86

The metric in Table 6 shows that the CNN-LSTM model performs exceedingly well compared to the RF regressor (Table 2). Moreover, ACD has performed better than the rest datatypes for MAE. The MAE value is the least for ACD datatype, similar to RF regressor model. Raw data has the most amount of error, as the noise level would be maximum for the data type. Next, the scoring functions will be discussed to see how far off the errors were underfitting the predictions repeated in this model too.

As can be seen in the Fig. 11 the predictions made by the model are exceedingly well and much better than RF regressor model. ACD predicted RULs are almost resembling to the groundtruth line and are even scored well (as seen in Table 7). In comparison to the raw data

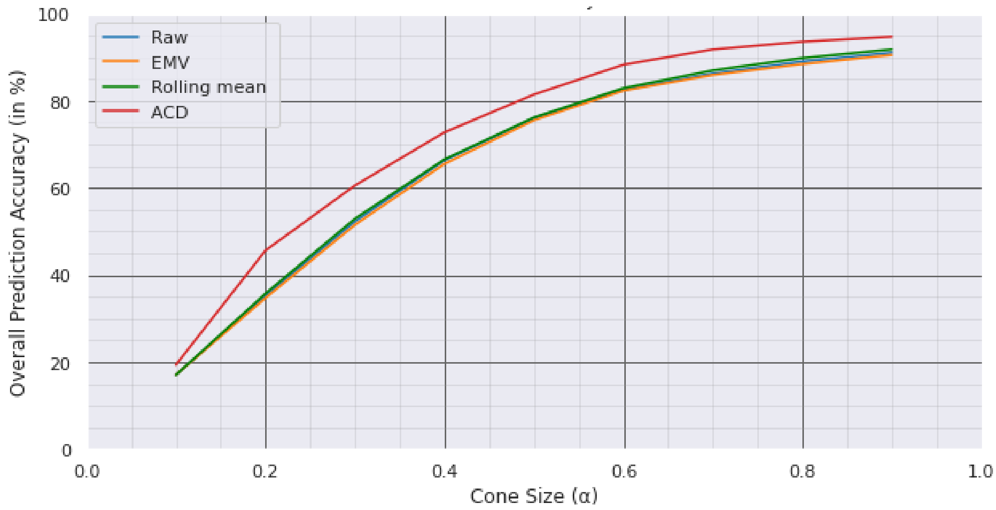

Figure 10. Overall prediction accuracy at varying cone size

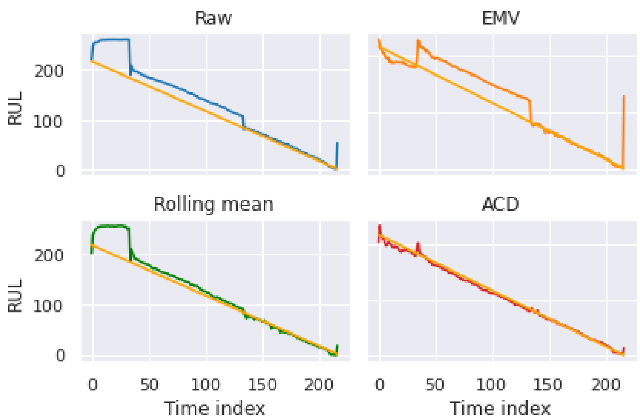

Figure 11. RUL prediction vs time for engine no. 19

The engine unit 19 is visualised with α-λ bounds in the Fig. 12 and the accuracy for each model is seen in Table 8. The accuracy has increased exponentially when compared with the RF regressor models. The models have predicted and shown better results. This is also seen in the overall prediction accuracy graph seen in Fig. 13. Each engine maintains its accuracy until the very last time interval. The accuracy is low in the end interval because the final RUL prediction is not yet perfected. The last interval shows that raw data performed better than the ACD model, however, the raw data was performing the worst since the first few intervals. The second best performce is seen from the ACD model.

type or the EMV, ACD rectifies the error and maintains its predictions throughout the time index. Therefore, the score value is also low and shows how it does not deviate much from the actual RUL.

Table 7. Scoring function and overall average score.

Data type	Average score no. 19 engine	Overall score for 20 test engines
Raw data	22.41	1326.45
EMV	87.52	318.22
Rolling mean	12.22	745.1
ACD	0.48	493.8

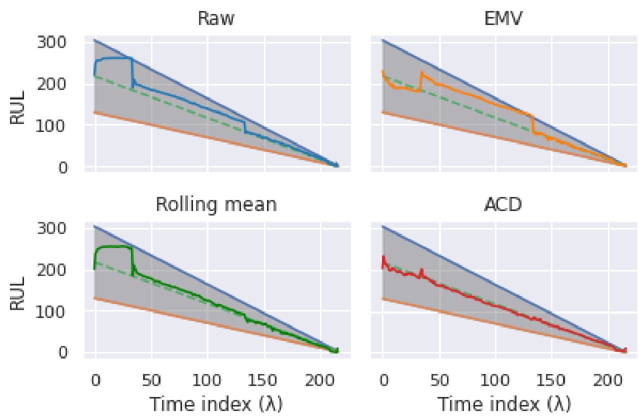

Figure 12. Alpha-lambda metric of engine unit no. 19 α = 0.3 and λ = 0.5

Similar to the first ML-model, the accuracy of the CNN-LSTM model will be evaluated with α-λ curves.

Lastly, the overall prediction accuracy for varying cone size can be seen in Fig. 14. The ACD has again performed well. However, other preprocessing methods (EMV and

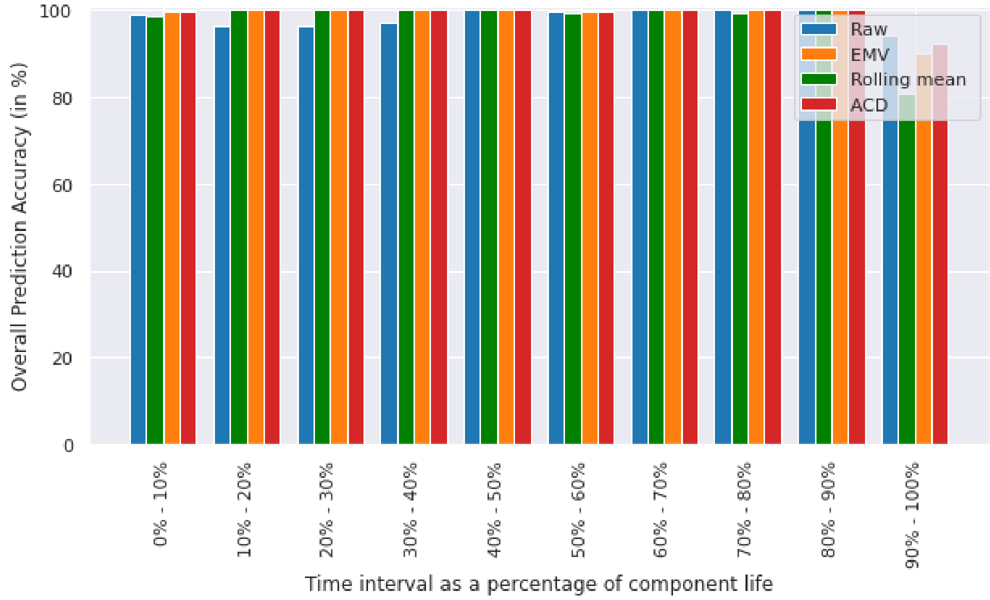

Figure 13. Overall prediction accuracy as a function of time intervals for each datatype. α = 0.3 and λ = 0.5

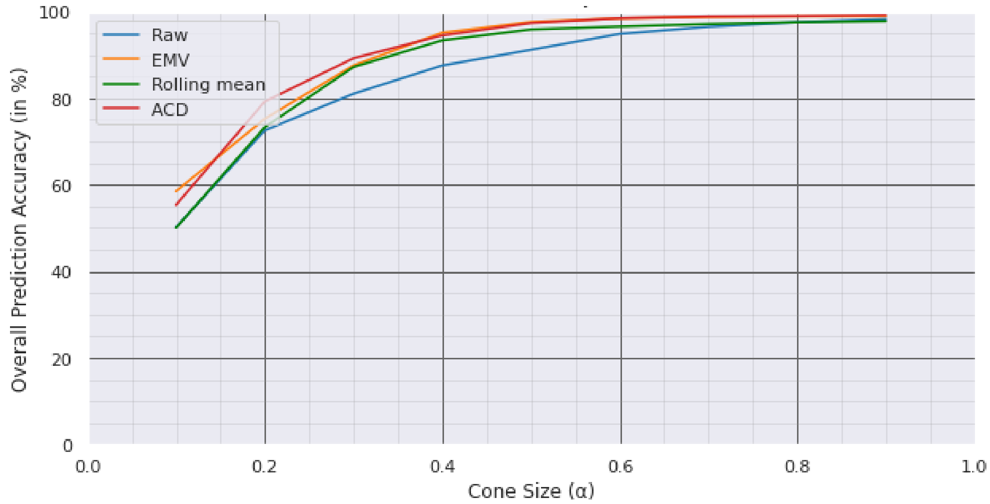

Figure 14. Overall prediction accuracy at varying cone size

Table 8. Overall accuracy of engine unit no. 19 α = 0.3 and λ = 0.5

Data type	Overall accuracy
Raw data	99.53 %
EMV	99.07 %
Rolling mean	96.77 %
ACD	99.53 %

Rolling mean) are very close to the accuracy of ACD. The EMV is closest to the ACD filter and it can be seen in the graph that it catches up from the α = 0.4. An interesting question arises here, what level of monotonic constraint is required for the best input to the model. Does it need to be entirely monotonic or can we modulate the monotonic constraint metric to achieve the best input. However, this is not the scope of the thesis and should be explored in the future.

Furthermore, comparing the two ML-models and their numerical results, the neural network performed significantly better, it can be compared with the Fig. 14 and Fig. 10. The accuracy has gone up for every data type with CNN-LSTM model. More improtantly, it also proves that using a monotonic constraint for fault progression data would improve the prediction and accuracy of the model but not signaficantly. A combination of monotonic constraint with CNN-LSTM would yield the better results. The impact of improving even smaller predictions increases the accuracy of RUL predictions and is on the cost of predictive maintenance and safety of flight. Accurate scheduling of maintenance activities could be called for and materials/components can have a longer life without hinderence to maintenance operations or the life of an aircraft.

VI. Conclusion

The proposal of the paper was to confirm if treating nonmontonic signals with monotonic constraints at preprocessing step in fault progression can improve accuray of predictions of RUL. The aim was successfully obtained by utilizing a monotonic constraint algorithm in preprocessing step that improved the prediction of RULs with high accuracy. The proposed ACD model was compared with other preprocessing methods for its performance and accuracy. Precise predictions were usually not possible, but having a range closer to the actual residual time was achievable with this method. Ultimately, improving the fault progression datasets by improving their quality of data to train and predict better. The proposed model would help the industry to avoid unwanted computing, increase lead time and improve predictions of RULs for the unwanted failure of components.

The tests carried out in the paper were to compare different preprocessing steps to enable the use of monotonic constraints. The nonmonotonic signal modulated to a monotonic signal by the ACD filter enabled machine learning models to learn and perform prognostics with higher accuracy. Most uncertainities occurred in the time cycle of a component's life were avoided while training the model. Estimating the trend's path of identifying RULs is much easier to interpret. As seen in the regressor and neural network model, the predicted error of the models was lowest for the monotonically constraint dataset, followed by rolling mean, exponential moving average, and the raw datasets. However, for the scoring functions, the overall average score was not the lowest for ACD model, which gave insights about its limitations. This causes the RUL predictions to overestimate the output and misalign from the groundtruth for a few cases. It causes underfitting in predictions. However, the prediction towards the end of the cycle is maintained and the final value is predicted closer to the groundtruth, which is of utmost importance.

Furthermore, ACD filter has its limitations while computing and forming monotonically constrained signals. The filter is applicable to only sensor values that are time-dependent and the sensor values have to be nonmonotonic in behaviour. The ACD filter has the ability to discard outliers and do not consider their properties while applying monotonic constraints. This could have However, the advantages of the ACD filter are very promising and have shown positive results.

In the future, it will be interesting to compare the proposed model with a monotonic regression model. The differences of preprocessing of monotonic constraints over a monotonic regression model would provide insights into the effectiveness of this process. Furthermore, complex datasets with more operating conditions could be experimented with this process. Even advancing into modulating monotonicity of data is a very interesting topic. As discussed before, the performance of ACD are very firm and the metric value is one. There wasn't a scope to modulate the metric value for this thesis. As it was seen, the EMV datatype (not fully monotonic) was accurately predicting close to the ACD datatype with the CNN-LSTM model. The optimal montonic constraint can be deduced from this experimentation.

A hybrid combination of physics-based model and the discussed data-based model will bring new possibilities for the industry of maintenance. Hybrid modelling are becoming popular and being experimented to bring the best of both worlds [37]. The data-based models can learn and improve exponentially with insights of the material properties and its physics.

Lastly, implementing monotonic constraint in phases of time cycles. Taking an example from the thesis, the accuracy of predictions was very low in the beginning of the time cycle for ACD datatype in RF regressor. If the monotonic constraints were not applied during the phase and only applied after the first few time cycles, an overall higher accuracy could be achieved.

References

[1] Lee, D., and Pan, R., "Predictive maintenance of complex system with multi-level reliability structure," *International Journal of Production Research*, Vol. 55, 2017, pp. 4785–4801. doi: 10.1080/00207543.2017.1299947.

[2] Brügge, F. A., "Predictive Maintenance Market The Evolution from Niche Topic to High ROI Application," `https://iotbusinessnews.com/2021/05/06/85410-predictive-maintenance-market-the-`

evolution-from-niche-topic-to-high-roi-application/, 2021.

[3] Frost, and Sullivan, "Global Airline Industry Predictive Maintenance Growth Opportunities," https://www.reportlinker.com/p06113016/Global-Airline-Industry-Predictive-Maintenance-Growth-Opportunities.html?utm_source=GNW, 2021.

[4] Sharma, A. B., Golubchik, L., and Govindan, R., "Sensor faults: Detection methods and prevalence in real-world datasets," *ACM Transactions on Sensor Networks*, Vol. 6, 2010. doi: 10.1145/1754414.1754419.

[5] Baalis Coble, J., and Baalis, J., "Trace: Tennessee Research and Creative Exchange Merging Data Sources to Predict Remaining Useful Life-An Automated Method to Identify Prognostic Parameters Recommended Citation," 2010. URL https://trace.tennessee.edu/utk{_}graddiss/683.

[6] Javed, K., Gouriveau, R., Zerhouni, N., and Nectoux, P., "Enabling health monitoring approach based on vibration data for accurate prognostics," *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 1, 2015, pp. 647–656. doi: 10.1109/TIE.2014.2327917.

[7] Wu Y, W. W., Hong G, "Prognosis of the probability of failure in tool condition monitoring applicationa time series based approach," *Int J Adv Manuf Technol*, Vol. 76, 2015, p. 513521.

[8] Huan Liu, M. D., Hiroshi Motoda, "A monotonic measure for optimal feature selection," https://link-springer-com.tudelft.idm.oclc.org/content/pdf/10.1007/BFb0026678.pdf, 2021.

[9] Niknam, S. A., Kobza, J., and Hines, J. W., "Techniques of trend analysis in degradation-based prognostics," *International Journal of Advanced Manufacturing Technology*, Vol. 88, No. 9-12, 2017, pp. 2429–2441. doi: 10.1007/s00170-016-8909-5.

[10] Bartoji, T., "Testing Monotonicity of Variables," 2010.

[11] Baraldi P, Z. E., Mangili F, "Investigation of uncertainty treatment capability of model-based and data-driven prognostic methods using simulated data," Vol. 112, 2013, p. 94108.

[12] Aizpurua JI, C. V., " Methodology for Design of Prognostic Systems," 2015.

[13] Feelders, A., " Monotone relabeling in ordinal classification," 2010, pp. 803–808.

[14] M. Rademaker, H. D. M., B. De Baets, " Optimal monotone relabelling of partially non-monotone ordinal data," *Optimization Methods and Software*, 2012, pp. 17–31.

[15] Jose-Ramon Cano, S. G., Julian Luengo, "Label Noise Filtering Techniques to Improve Monotonic Classification," 2018.

[16] I. Milstein, R. P., A. Ben-David, "Generating noisy monotone ordinaldatasets," *Artificial Intelligence Research*, Vol. 3, 2014, pp. 30–37.

[17] Kvaloy J T Lindqvist B and Malmedal H, "A statistical test for monotonic and non-monotonic trend in repairable systems," *Proc European Conference on Safety and Reliability ESREL 2001 Torino 2002*, , No. May 2014, 2002, pp. 1563–1570.

[18] Daniels, H. A. M., and Kamp, B., "Application of MLP Networks to Bond rating and House Pricing," *Neural Computing Applications*, Vol. 8, 1999, pp. 226–234.

[19] Sill, J., "Advances in Neural Information Processing Systems (NIPS)," Vol. 10, 1998, pp. 661–667.

[20] Ben-David, S.-L., A., and Tran, T., "Adding monotonicity to learning algorithms may impair their accuracy," *Expert Systems with Applications*, Vol. 36, 1999, pp. 6627–6634.

[21] R. Potharst, J. B., "Decision trees for ordinal classification," 2000, pp. 97–111.

[22] Potharst, R., and Feelders, A. J., "Classification trees for problems with monotonicity constraints," *ACM SIGKDD Explorations Newsletter*, Vol. 4, No. 1, 2002, pp. 1–10. doi: 10.1145/568574.568577.

[23] H. Daniels, M. V., "Monotone and partially monotone neural networks," 2010, p. 906917.

[24] H. Zhu, X.-Z. W. R. A. R. A., E. C. Tsang, "Monotonic classification extreme learning machine," 2017, p. 205213.

[25] David, B., "Monotonicity Maintenance in Information-Theoretic Machine Learning Algorithms," *Machine Learning*, 1995, pp. 29–43.

[26] Gale A, M. A., "Explaining Monotonic Ranking functions," Vol. 14, 2020, pp. 640–652. doi: 10.14778/3436905.3436922.

[27] Velikova M, D. H., "Enforcing monotonicity of decision models: algorithm and performance," 2003.

[28] N., B., "Learning monotonic models from data," 2013.

[29] Brownlee, J., https://machinelearningmastery.com/moving-average-smoothing-for-time-series-forecasting-python/, 2016.

[30] Vamo, C., "Automatic algorithm for monotone trend removal," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, Vol. 75, No. 3, 2007, p. 036705. doi: 10.1103/PhysRevE.75.036705, URL https://journals-aps-org.tudelft.idm.oclc.org/pre/abstract/10.1103/PhysRevE.75.036705.

[31] Vamo, C., and Maria, C., "Automatic trend estimation," 2013. doi: 10.1007/978-94-007-4825-5, URL http://link.springer.com/10.1007/978-94-007-4825-5.

[32] Saxena A., G. K., "Turbofan Engine Degradation Simulation Data Set," *NASA Ames Prognostics Data Repository*, 2008.

[33] Jayasinghe, L., Samarasinghe, T., Yuen, C., and Ge, S. S., "Temporal Convolutional Memory Networks for Remaining Useful Life Estimation of Industrial Machinery," *arXiv preprint arXiv:1810.05644*, 2018.

[34] Acharya, S., https://towardsdatascience.com/what-are-rmse-and-mae-e405ce230383, 2021.

[35] Goebel, K., Saxena, A., Saha, S., Saha, B., and Celaya, J., "Prognostic Performance Metrics," *Machine Learning and Knowledge Discovery for Engineering Systems Health Management*, Vol. 22, 2011, p. 147. doi: 10.1201/b11580-

7.

[36] Breiman, L., “Random Forests,” *Machine Learning*, 2001, pp. 5–32. doi: 10.1023/A:1010933404324.

[37] Nikolaev, S., Belov, S., Gusev, M., and Uzhinsky, I., “Hybrid Data-Driven and Physics-Based Modelling for Prescriptive Maintenance of Gas-Turbine Power Plant,” *IFIP Advances in Information and Communication Technology*, Vol. 565 IFIP, 2019, pp. 379–388. doi: 10.1007/978-3-030-42250-9$_3$6.

II

Literature Study

1. Introduction

The scheduled regulatory aircraft maintenance are performed periodically and it serves the purpose of monitoring components of the aircraft. However, unexpected failures or unanticipated breakdowns of components cause disruption in operations, the cause of safety, and eventually affect the cost structure of the airline [34]. Preventive maintenance was performed for the longest time in the past and eventually the industry diverted towards predictive maintenance for its cost benefits and effective operations [15]. Operations of maintenance of aircrafts are very complex and require attention to detail [26]. Constant research and experiments are carried out to perfect the prognostics and perform predictive maintenance robustly and effectively. This research aims to add further to improve the predictive maintenance for degrading components of an aircraft.

The prognostics and health management (PHM) have multiple approaches to perform prognostics and schedule accurate maintenance. In this research, the data-driven approach is opted for its cost-effective benefits, growing interest, and popularity in the industry. The materials and components that degrade overtime theoretically have a logarithmic behavior in their sensor data. Due to the noise and disturbances processed by the sensors of the component, monotonic trends are not obtained. These uncertainties cause failure in prediction as the ML-models possess unwanted extra information. Therefore, through this literature study, the author explores prognostics and used cases to prepare and develop a plan to execute this research effectively.

The literature review explores multiple topics: predictive maintenance, Prognostics and health management (PHM), Remaining Useful life (RUL), Machine learning (ML), non-monotonic and monotonic signals, monotonic improvement techniques, and evaluating the monotonic signals and their performances. The conceptual research model is presented with the research questions and objectives in Section 3. The methodology and experimental setup are discussed in Section 4. Finally, the work break-down is explained in a Gantt chart.

2. Literature Review

In this section, the growth of predictive maintenance in aviation are discussed. The current prognostics methods and the data-driven methods used to estimate RUL are studied. Machine learning models are presented that are used on a regular basis. The section deep dives into monotonicity and its methods for treating non-monotonic signals. The used cases for monotonic metric and monotonic constraints are presented. Lastly, the dataset to be experimented with is discussed.

2.1. Predictive maintenance in Aviation

The increasing aviation population results in demand for more aircraft. More aircraft result in growing complexity in each department of operations of an airline. Specifically in the scheduling department. The planning and execution of multiple fleets and crew requires an organized system for smooth functioning of the airline company. Similarly, aircraft are maintained and checked periodically to avoid any harm to the aircraft and the human life on board. For economic and safety purposes, the maintenance activities of an aircraft must be robust [18].

The scheduled regulatory maintenance is known to perform periodically and it serves the purpose of checking up and monitoring components of the aircraft. However, the unexpected failure maintenance or unanticipated breakdowns cause disruption in operations and affect the airline heavily on cost and safety. The airline industry used preventive maintenance for an extended period of time to avoid such unwanted failures. It works best when there is a solid relationship between equipment age and failure rate. For example, when a material property changes due to fatigue. The probability of failure can be estimated and a maintenance can be scheduled before breakdown [18][36]. For the health of the system, the analytical solutions are not able to predict failures but provide an alert with sensor data as a baseline indicator [71]. Therefore, it increases the initial costs and requires frequent access to equipment [44].

Predictive maintenance prevents unwanted failure by keeping a constant watch on the system and providing alerts with condition monitoring [36]. Data collection and analyses programs could be substantial, but the recovery and savings would offset the costs easily [44].

Predictive maintenance promotes prognostics based life extension and has been effective compared to other models. The aerospace industry is the most vibrant research and development activity in systems prognostics.

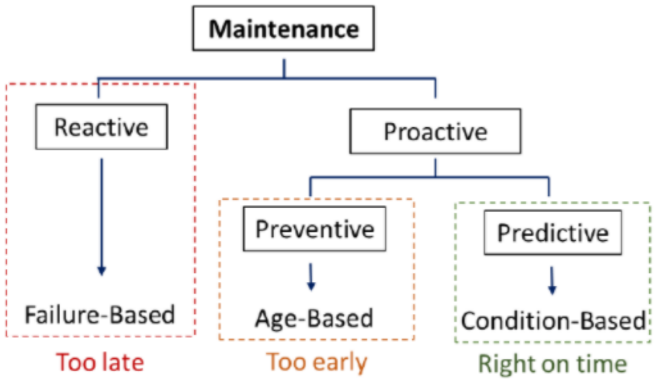

Figure 1: Difference in reactive, preventive and predictive [18]

The algorithms monitor aircraft structures, avionics, propulsion systems, etc, and their functionality is embedded into the health management system to reduce life-cycle cost and improve flight readiness [57].

PHM- Prognostics and Health Management

The goal of a PHM is to have a reduction of operation and support costs while maintaining or increasing the availability of systems in an industry. "PHM is an approach to system life-cycle support that seeks to reduce or eliminate inspection and time-based maintenance through accurate monitoring, incipient fault detection, and prediction of impending faults" [14][59]. It ensures detecting and informing system faults for maintenance. Along with fault detection and fault isolation, the PHM detects fault prediction on selected components, fault filtering and reporting, predicting Remaining Useful Life (RUL), health management, and recommended actions to pilot when necessary [27]. It is important for PHM systems to obtain the actual time condition information of the subsystems. For example, to predict RUL of a system, the PHM system must initially combine the interpretation of the environment. After which, the operational and performance parameters are deduced to assess the health of the product [14].

According to [51], PHM consists of the following features: Raw data, Diagnostics, Prognostics, and health management. The model is designed by Sandia National Laboratories (SNL), which is shown in the paper [51] Figure A.1. The evidence engine aims for feature extraction, trend detection, and estimation of RUL, whereas the consequence engine can analyze the end result of the maintenance action [14]. This thesis will explore the evidence engine and specifically the estimation of RUL.

Remaining Useful Life

The remaining useful life of a subsystem can be analyzed by using sensor data and using the prognostics technique [2]. RULs provide decision makers with information about the health of the component which would allow them to make repairs or changes in the regular operational characteristics (like load) and eventually increase the life cycle of the component. The estimation of RUL helps planners to transition smoothly from faulty equipment to a fully functional one by informing the upcoming maintenance [58]. Algorithms of prognostics predicting RUL are classified as degradation-based or Type III prognostics [16].

The degradation-based modeling of a system or component is defined as the length from the current time to the end of its useful life and it can be used to characterize the system current health status [13]. From Figure 1, at (t_0) current time, the RUL estimation shows that the health of the component would deteriorate until the decided threshold level, that is when the component reaches t_1. The difference between the two points is the RUL estimation of the component at t_0. As the estimations are not very accurate due to uncertainty, the probability distribution function (pdf) is approximated for the precision of prediction [13].

RUL is further classified into three different types of methods based on [13]. The model-based approach for RUL merges the system behavior and the measured data. Types of models in this approach are physical failure models, stochastic filtering models, and statistical models. The advantage of model-based approaches is the ability to understand the physical fundamentals of the monitored system. Hence, it would show higher

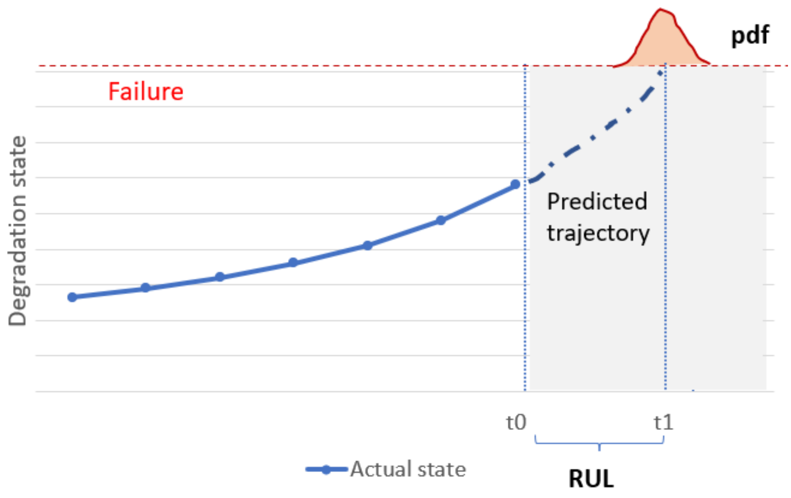

Figure 2: Estimating RUL of a Degrading component from its signal values from the data[13]

accuracy than a data-driven model. However, it is more often costly, complex, and difficult to achieve the accuracy required for the system [13].

Data-driven approaches, on the other hand, use past recorded data to analyse and then predict future. It does not incorporate physical models to estimate RUL. The only source of understanding of degradation is through the measured input/output data. The most frequently used models of the approach are machine learning methods and graphical models. A graphical model represents and indicates the conditional independence between the random variables. Whereas machine learning methods train samples and identify the cross-dependencies between the variables [13].

The accuracy of the approach depends on the quality and quantity of the data. The advantage of the approach is to convert the complex noisy data into usable and understandable data for the prognostics and estimation of RUL. "The process of traceability and capitalization of data is a key element in the context of the evolution of the maintenance towards predictive strategies" [23]. The disadvantage is that it is not easy to apply data-driven approaches because there is no specific or efficient procedure to obtain training data [13][74].

Lastly, the fusion approach, combination of the prior two approaches establish an effective way to overcome the limitation of each method and predict the RUL with an improved accuracy. The method for the approach is that the first one would estimate RUL and improve accuracy. The second part is where each approach predicts the RUL individually and they are combined with probabilistic methods to obtain a better predicted RUL [13].

For the scope of the research, the data-driven approach would be further explored. Machine learning methods are going to be adopted for the process and therefore will be discussed in brief in the following.

2.2. Machine learning

Machine learning is a tool of optimizing methods that are used to train models with collected data and predict the future based on the information known [9]. The system can be used for cautioning the user and making aware of the possibilities that can be experienced [47].

The data sets available for modeling are divided into training, testing, validating data set. Training set is used for fitting the different models and making the model work, the validation set helps to evaluate model selection and estimate predictive performance. Testing data is the data that the model has never seen before, hence it helps in evaluating the output. The more the available data, the better quality will be the models [47]. There are four stages in machine learning methods which require human intelligence. Out of these stages, the most time consuming and resource driven are the data collection and optimization stages [9].

Machine learning group

Machine learning algorithms are organized into groups: supervised, unsupervised, and reinforcement learning [47].

Supervised learning closely experiences with the help of labels given already. The purpose of the labels in the data aids the algorithm to correlate their features [47]. So, a training data x has label y. Labeling the data helps the algorithm to automate complex tasks. Therefore, reducing the loss function by training a model by labeling data (with the Equation 1). Hence, Root mean square error (RMSE) is one of the considering factors for evaluating the results of the model. The availability of an extensive and high-quality labeled data set is crucial and can enable the training of even a deep convolutional neural network. The two common tasks applications of the group are classification and regression [9].

$$L = ||y - \phi(x;\theta)|| \tag{1}$$

Classification: If the labels are discrete, then the task is classification. Applications of classification could be recognized if the image is of a dog or cat, the stock market would rise or fall tomorrow, or a component would fail before a particular day or not [47][9].

Regression: If the labels are continuous, then the task is regression. Application of regression includes predicting sales for a new product or a lift profile for an airfoil shape [47][9].

In **unsupervised learning**, or data mining or pattern extraction from data are used when there are unclassified and unlabeled data. The clustering of common tasks to group similar examples together enables more insights into the data and the correlations and dependency existing between their features. Three common techniques of clustering are k-means, mixture model, and hierarchical clustering [8][70]. To find a low-dimensional subspace, parameterized by a latent variable z, which shows a high-dimensional state x, a goal could be set to find two functions, encoder $z = \phi$ and a decoder $\hat{\mathbf{x}} = \psi(\mathbf{z})$, so that $\hat{\mathbf{x}} = \psi(\phi(\mathbf{x})) \approx \mathbf{x}$ [9]. The functions ϕ and ψ are implicitly parameterized by weights θ that must be tuned to minimize the following loss function (seen in Equation 2 [9]. When a encoder and decoder are linear functions, then the optimal embedding recovers the classical singular value decomposition (SVD) or principal component decomposition (PCA) [8][10].

$$L = ||x - \psi(\phi(x))|| \tag{2}$$

Reinforcement learning (RL) is related to goal-oriented algorithms. It has the ability to learn from interactions with the environment [63]. The method allows the algorithm to automatically identify the natural behavior of the environment and contextualize it to improve performance by taking decisions [47]. RL agents are capable of learning challenging delayed rewards. An example of a delayed reward would be a self-driving car that can only be rewarded if it completes its task of reaching the final destination safely. "Simple reward feedback is required for the agent to learn which action is the best and this is known as the reinforcement signal" [47].

Deep Learning is based on neural networks and is one another type of ML models. Neural networks (NN) are powerful because of the vivid expressive representation of data and their diverse architecture [33]. NNs require a large amount of data for processing the models. However, it also important to know that they tend to overfit to data [9].

Data mining

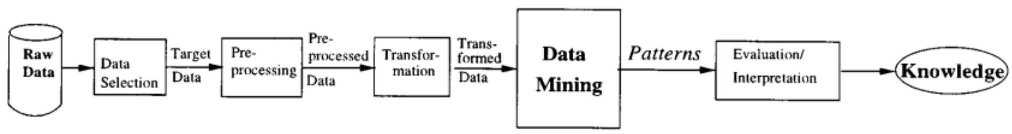

Figure 3: KDD process [43]

Knowledge about the database is of utmost importance. Data mining (DM) is a very powerful tool which identifies with automated data analysis. The DM occurs in the process of Knowledge discovery in databases (KDD) as seen in the Figure 3. To ensure the right data is passed on to data mining, the steps before DM have to be taken. DM is desired because it increases the ease of collecting data over networks with reduced cost, robust ML algorithms are developed to process this data, and lastly, it enables the use of computationally intensive methods [43][45].

Neural Networks

According to [43], for the longest time, soft computing methodologies like neural networks did not suffice the data mining criteria because of their black-box nature [64]. However, with recent progress in extracting embedded knowledge in trained networks, neural networks have an advantage over other machine learning algorithms in terms of scaling. Neural networks are indeed very suitable in data-rich environments. They are used for quantitative evaluation, clustering, classification, and regression.

Type of models for data mining

The author of [43], classifies different models used for various situations.

1. Classification [42][3]: A data item is grouped into several predefined categorical classes. It is the most simple binary classification. There can be multiple categorical classes in multi class classification.

2. Regression [19][22]: Focuses on displaying a sensor data to a real valued prediction variable. This one predicts variable and not the class.

3. Clustering [68][35]: A grouping of data sets depending on on their metrics and probability density models are called as clusters. It sets each data item into these clusters.

4. Sequence analysis[43]: The time series data set results in patterns which are sequential. The idea is to use these sequences for further extraction and deviation reporting.

Under-fitting and over fitting

Under-fitting The bias of such models is very high. The reason is because the model poses fewer features and therefore cannot learn from the data very well [47].

Over-fitting The variance of the model is very high. The model has complex functions for fitting to the data accurately but not generalized enough to be able to learn from trends for predicting new data. The Figure 4 describes the generalization error, bias, and optimal capacity for a model to be able to fit just enough for a robust fit [47]. The test set data of machine learning is used to avoid the overfitting of model. Over fitting as a phenomenon can be controlled by addressing a few options. Firstly, reduce the number of features. Make decisions on why to remove the selected feature. Secondly, regularization is useful with a lot of useful features. Lastly, early stopping can be used while training the learning algorithm iterative (as seen in Figure 5) [47].

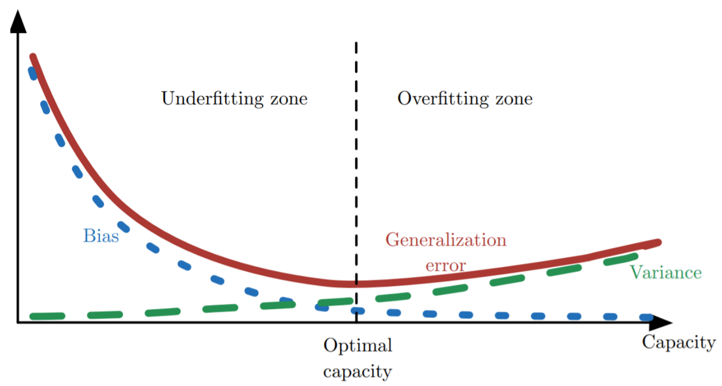

Figure 4: Bias and Variance [33].

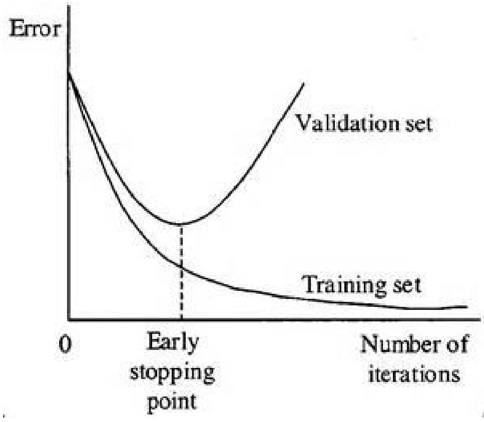

Figure 5: Early stopping feature to avoid over-fitting [47].

2.3. Monotonicity vs Non-monotonicity

In supervised learning, the assumption on the functional relation between a feature space and target space is approximated. Otherwise, the inductive learning task cannot be estimated because not enough domain knowledge is present. However, in many applications, prior knowledge is frequently available.

These applications could be represented by monotonicity when there is an increasing or decreasing relation between the target variable and the predictor variable. Hence, monotonic behavior represents as a good feature for prognostics [50]. This section shows insights into the topic of (non)monotonicity in machine learning and provides a monotonic metric for identifying and analyzing results.

Monotonic constraint poses a monotonic trend where the condition is fulfilled if and only if the trend is either entirely nonincreasing or entirely non-decreasing [50]. The degradation of material is irreversible and eventually the component's state replaces a healthy to an unhealthy state. Which suggests that a feature of the component or system would show a degrading trend in the data. Therefore, changing monotonously in the decreasing propagation as illustrated in Figure 6 [50][37]. However, feature trends have noisy data and it is difficult to extract the right data for this method to be accurate [16]. Furthermore, a monotonic fault progression has been considered as a vital assumption for a lot of prognostics models [16][46][61][48]. Therefore, non-monotonic degradation features increase uncertainty in machine learning models compared to a monotonic unidirectional degradation data set. The Figure 6' hypothesis is the output of monotonic trend vs nonmonotonic trend. As the fault progression would be monotonic, the window of pdf would be smaller compared to the nonmontonic output. Therefore, we will be able to predict the RUL accurately. This hypothesis will experimented in this thesis.

To treat non-monotonic data, few steps are required. Firstly, a metric that calculates monotonicity has to be established for each feature trend. Furthermore, a method to analyze the effect on feature trends of monotonic metric due to noise level. As the computation of monotonicity changes when the noise level increases [50]. Finally, a model to treat the trends which will achieve monotonicity. "Imposing monotonicity acts as a regularizer, improves generalization to test data, and makes the end-to-end model more interpretable, debuggable, and trustworthy" [72].

The impact of change points and regime shift leads to the existence of non-monotonic degradation. A prognostic capable system improves maintenance and one can expect a result of non-monotonic prognostic parameters upon post-prognostic maintenance. The prefailure repairs caused due to human intervention or self-healing are also an example of the same (fatigue crack closure) [48][40]. These non-monotonic degradation appears in the modeling process and increase the uncertainty of the prognostics and disrupt the failure predictions [75]. These degradation parameters need to be assessed and a monotonic metric needs to be evaluated first.

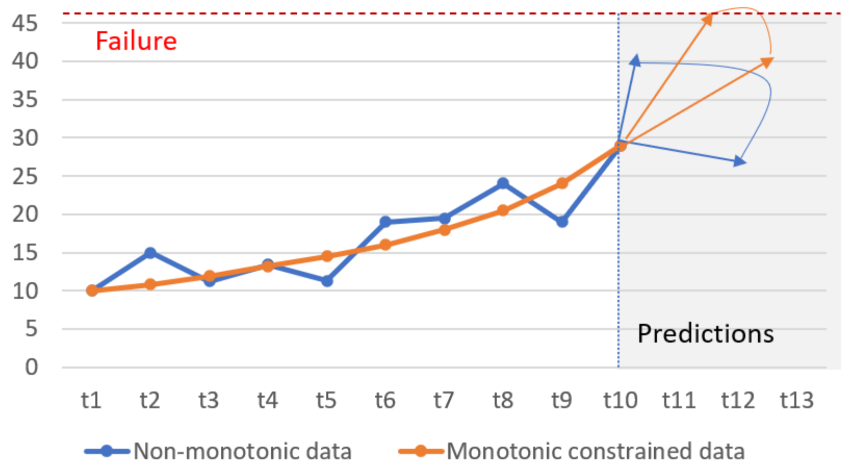

Figure 6: Trends of entirely non-decreasing and non-increasing functions.

2.4. Non-monotonic trends in components

In the aviation industry, maintenance checks are carried out on different physical parts and technical software checks too. The body of the aircraft, engines, actuators, etc are all maintained regularly. In today's time, aircraft are wired with multiple sensors around the aircraft. It improves the task of a maintenance engineer dramatically. With more sensor data, a better check could be kept on aircraft health.

These individual sensor outputs do not necessarily provide the health situation of a component of the aircraft. Some require to be in combination with others to make sense of the degradation patterns of the component. As established in the previous section, it is vital to have the data monotonic in terms of propagation, to have a good prediction. The sensors that show non-monotonic behavior can be treated with algorithms to improve the accuracy. However, it is important to first identify these parameters which are non-monotonic. In this research, the data set of a 90k turbofan engine from C-MAPSS is considered (discussed in section 2).

Therefore, the different 21 sensor data have to be individually analyzed by graphing them and using the monotonic metric (from subsection 2.5) to check their monotonicity. Evaluation on the requirement of an extreme non-monotonic sensor can be then decided based on modeling of the algorithm. If extreme outliers occur in the final health management output, further analysis has to be done on the sensor data.

2.5. Trend detection for monotonicity and noise reduction

"The quality of feature parameters extracted from the monitoring signal determines the complexity degree of prognostic methods."[73] The system degradation progression can be evaluated by estimating the feature quality with monitoring monotonicity. Following are a few methods to compute a monotonic metric for finding out the best health indicators or features for prognostics

Statistical outlier detection techniques

The author of [6], proposes a method for monotonic decision trees via information-theoretic top-down induction decision tree (TDIDT) algorithm which uses entropy for attribute selection. This method does not guarantee to be effective as the non-monotonic trends are not outliers.

Local Gradient Based (LGB) method

Author of [16] describes, "The average difference of the fraction of positive and negative derivatives of the time series of a feature over time." The idea is to smooth the data to give accurate estimates, but the noise makes the method impractical and inaccurate. The formula used to compute the monotonicity metric is given by Equation 3. Where G^{+} and G^{-} represent vector positive and negative derivatives respectively (seen in Equation 3).

$$\mathrm{MM}_1 = \mathrm{mean}\left(\left|\frac{\#[G^{+}]}{N-1} - \frac{\#[G^{-}]}{N-1}\right|\right) \tag{3}$$

with

$$G_i^+ = \frac{\Delta y_i^+}{\Delta t_i} = \frac{y_i - y_{i-1}}{\Delta t} \mid (y_i - y_{i-1}) > 0$$
$$G_i^- = \frac{\Delta y_i^-}{\Delta t_i} = \frac{y_i - y_{i-1}}{\Delta t} \mid (y_i - y_{i-1}) < 0 \tag{4}$$

The same formula can be rewritten in the following Equation 4[16]

$$\text{Monotonicity} = \frac{1}{M}\sum_{j=1}^{M}\left|\sum_{k=1}^{N_j-1}\frac{\operatorname{sgn}\left(x_j(k+1) - x_j(k)\right)}{N_j - 1}\right| \tag{5}$$

Improved Separability/RMI Based Method

Separability index s_k, obtained by segmenting time-series features into equal distances, as shown in the Figure 7. The d is distance between 25th and 75th percentile at window index k and a is non-overlapping in consecutive window frames. After calculating the separability indices, performance of the feature is calculated with the Equation 6 and defined as the average of separability indices s_k [50][73][11].

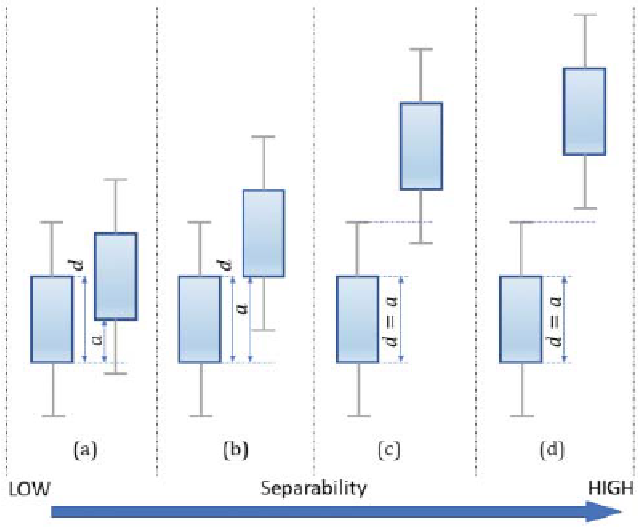

Figure 7: Visual representation of feature separation [50].

$$\text{MM}_2 = \frac{1}{M}\Sigma_{k=1}^{M} s_k \tag{6}$$

Furthermore, a ranking mutual Information (RMI), derived from Shannon entropy is a measure of classification consistency. The technique is proposed from [28]. It is the index between two rankings of a random variable. A combination of the two methods can evaluate features and even select features in case of ordinal classification [28].

Euclidean distance and correlation coefficient

The same steps as the previous model would be carried out, but the separable distance s_k is replaced with Euclidean distance and RMI is replaced by Pearson's correlation coefficient. The improved RMI reduces a significant amount of noise. The Euclidean distance D_k between two vectors is given by Equation 7. The values are based on the comparison between 75th percentile of two successive segmented feature vectors. The flow chart and step can be seen in the Figure 8 and further information on the procedure can be found in [50].

$$D_k(X_k, X_{k-1}) = \sqrt{\sum_{j=1}^{L}\left(x_j^k - x_j^{k-1}\right)^2} \tag{7}$$

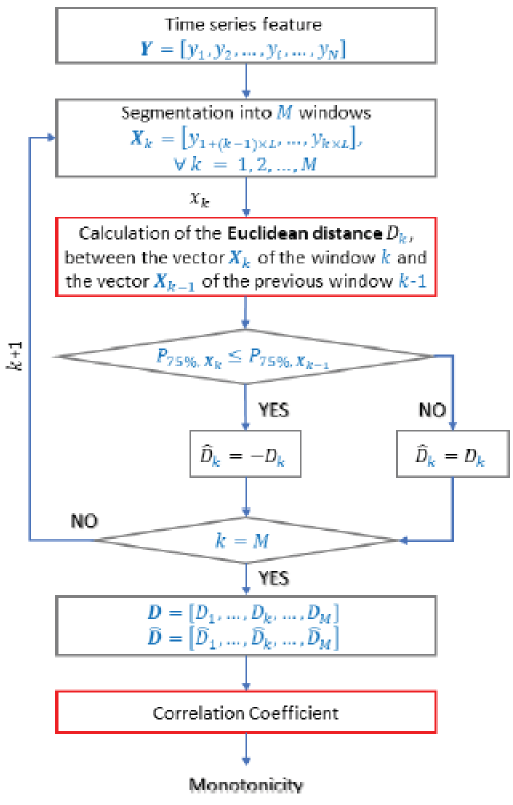

Figure 8: Calculation flowchart for the improved euclidean distance and Pearson's correlation [50].

2.6. Case studies

The prediction of RUL in degradation signals has been tested with several state-of-art algorithms. To improve on the accuracy of the predictions, these are methods and models performed in the past.

Feature extraction using trigonometric and cumulative function

In the paper of [31], the author preprocessed the data for feature selection with three characteristics, namely, monotonicity, trendability, and predictability. The features were extracted by trigonometric and cumulative functions, which help to transform the raw data into selective features which improve the long-term prognostics. The following steps suggested were to use the Summation Wavelet Extreme Learning Machine (SW-ELM) for improving prediction performance. For further improvement, an unsupervised classification approach is adopted, Subtractive Maximum Entropy Fuzzy Clustering (S-MEFC). Maximum entropy inference is used to identify the uncertain unlabeled data and label the number of states. The paper was successful in setting failure thresholds and estimating RUL.

Synthesized Structural health prognostics for identifying health conditions

The paper [69], proposes a generic health index system framework for structural health prognostics. It includes Synthesized health index technique (SHI) for enabling heterogeneous sensory signals; an offline learning scheme using Sparse Bayes Learning (SBL) technique for enabling kernel functions in real-time for RUL; an online learning scheme using Similarity-based interpolation (SBI) for predicting RUL with background health knowledge and a map for managing the prognostic's uncertainty for involving the statistical characteristics of the RUL. Regardless of the complexity and size of the data, the SHI health index proved to define the degree of health conditions. Two case studies performed with the method showed promising results .

Deep convolution neural network based regression

A convolution neural network was used for computer vision tasks or natural language processing in the [2] paper. The authors used the pooling filter technique for automation of multichannel sensor raw data for feature learning systematically. It helped to learn the salient features automatically. The data set used for experiments was the NASA C-MAPSS data set and PHM 2008 data challenge data set. The scoring function (S) is given by Equation 8. Compared to RMSE, this function considers the actual risk of estimating RUL. It disciplines late predictions compared to early predictions. However, there are a few issues with this method. Firstly, A single outlier with late prediction dominates the overall score. Which can be misleading as there are outliers in the data now and then. Secondly, this function would prefer the algorithms that lower the score by underestimating RUL.

$$S = \begin{cases} \sum_{i=1}^{N} \left(e^{-\frac{h_i}{13}} - 1 \right) & \text{for } h_i < 0 \\ \sum_{i=1}^{N} \left(e^{\frac{h_i}{10}} - 1 \right) & \text{for } h_i \geq 0 \end{cases} \tag{8}$$

Multivariable prognostic models

These types of models with techniques that are not polished, usually show worse performance to fit the data set into the model and eventually predict inaccurate results. The authors of [25], discuss an interpretable index of predictive discrimination and methods for identifying the calibration of predicted survival probabilities. They also elaborate on issues with a poorly fitted regression model. The paper also warned about the pitfalls while modeling a multivariable prognostic model.

Monotonic constraining methods

Principal component analysis (PCA)

PCA is primarily used for analyzing features. PCA was performed in [49] for multivariate trend analysis, along with Independent Component Analysis (ICA) for degradation signals. The goal of PCA is to analyze the covariance structure and reduce the complexity of data. In [49], PCA smoothed acoustic emission signals by eliminating highly correlated variables. However, it was not able to entirely convert the signals into complete monotonic behavior.

Average conditional displacement (ACD)

This algorithm is used to automatically estimate the monotonic trends. The algorithm is based on a signal value interval, but it works fairly well for estimating the monotonic trends of a stationary noise-filled time series data [48]. The algorithm results in linear estimations which have a slope proportional to the average of time series values. The two advantages of the algorithm are that no initial assumptions of the trend are required and the algorithm is developed automatically [65]. The autoreffig:acd shows the variation of time series in one step indicated with the thick line.

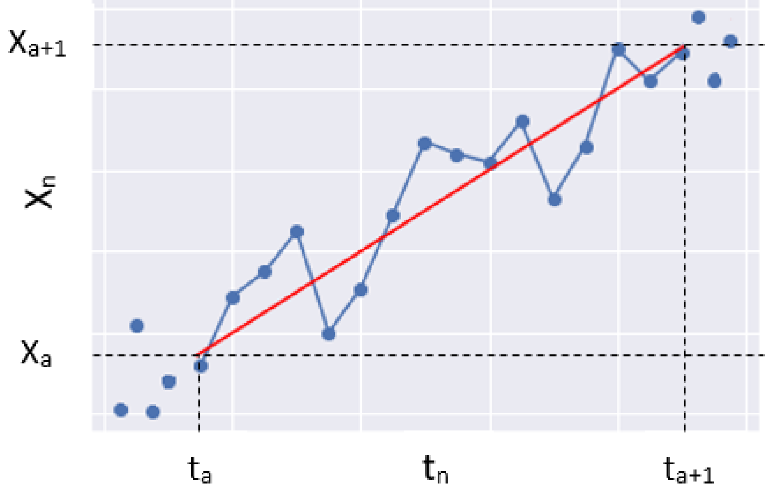

Figure 9: ACD approximation in an interval [48]

Monotonicity of tree-based methods

A monotonic trend could be splitted when tree-based methods like random forest or decision trees are used to fit the data set as seen in the Figure 10. Here the red line tries to fit the data and fails to give a monotonic trend. By using monotonicity constraints in LighGBM and XGBoost algorithms, it is possible to generate the monotonic trends as seen in Figure 11 [52][17]. However, there is noise existing in the graph and the signal is not smooth enough for to compute prognostics.

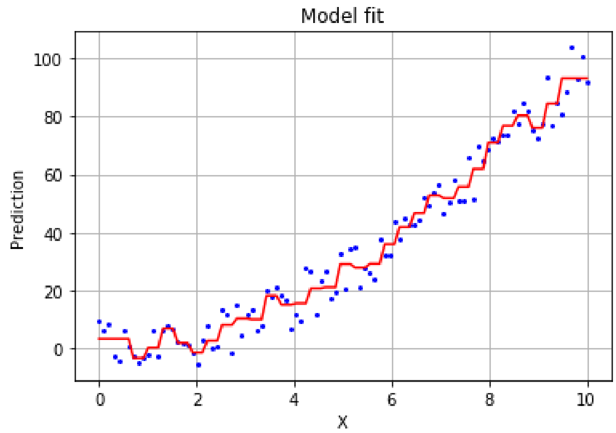

Figure 10: Model with non-monotonicity constraint algorithms[17]

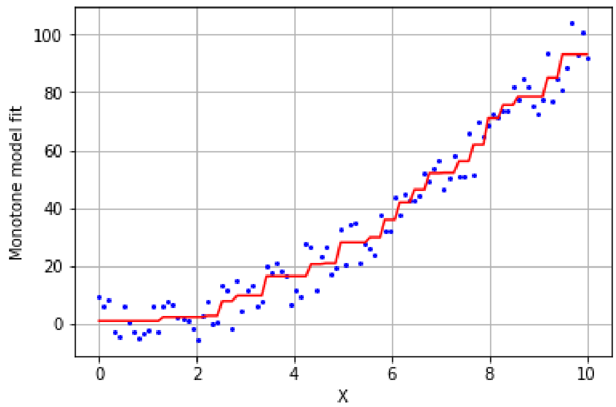

Figure 11: LighGBM and XGBoost algorithms used on the data set[17]

2.7. Dataset chosen for experimentation

The author chooses to develop a process that improves on the accuracy of prognostics of degradation of the component, it is essential to experiment with the data that have the final results (RULs in this case) for approval and understanding. Testing for such data is selected that replicates the real life scenarios and enables the algorithm to be used in a generalized format. Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) is a simulated data set used for replicating real-life commercial turbofan engines. The description of the engine model 90K is given below [21]:

- 90,000 lb thrust which is 400,340 N
- Atmospheric model altitude from sea level to 40,000 ft,
- Mach 0 to 0.9,
- Sea-level temperature -60 to 103 F

Engine control system [21]:

- Throttle-resolver angle (TRA) specific to a fan speed controller
- Three high-limit regulators prevented from exceeding its threshold limit for core speed, engine pressure ratio, High-pressure turbine (HPT) exit temperature
- Regulator to avoid static pressure at high pressure compressor (HPC) exit from going too low
- Moreover, to control the core speed, acceleration and deceleration limiters also exist

The data set inputs, output, and equilibrium values for all flight conditions are given in the Appendix A. The simplified turbo engine design is represented in Figure A.2 and the workflow of the engine with ducts and bleed omitted is seen in Figure A.3.

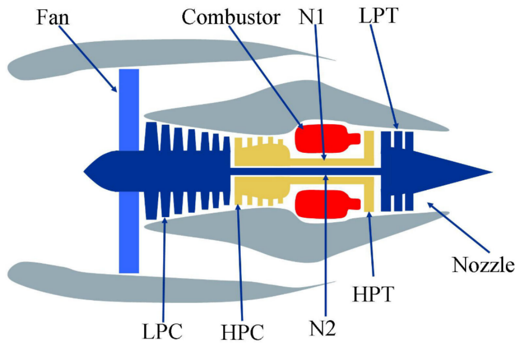

Figure 12: A 90k turbofan engine, simplified version[21]

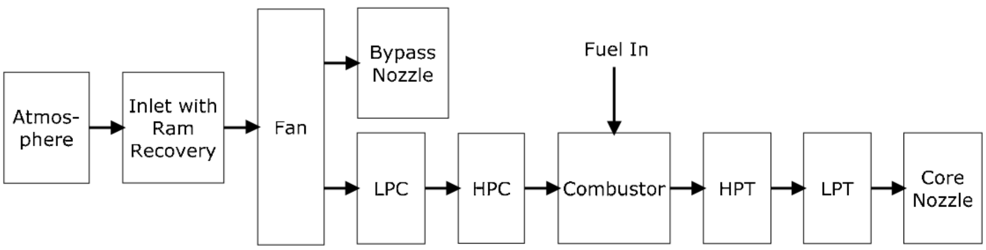

Figure 13: Working of the engine divided in simulation boxes [21]

Data-set characteristics

The purpose of selecting the C-MAPSS data set was because it has varied characteristics and are very helpful to compute realistic prognostic models. These publications do not focus on the physics-of-failure of turbofan engines but describe the generation of these data sets and various practical aspects when using C-MAPSS data sets for prognostics. Characteristic of the data set is provided from [54][58] as follows:

1. To replicate a real system of the aircraft engine, a data of non-linear system of high fidelity is created for a multi-dimensional response
2. The data consist high level of noise
3. Fault conditions are in-cooperated in operational systems
4. Data of different level of complexities are given for training and testing the algorithms. The description of the data-sets is given in the Figure A.4

5. The data sets number 1-4 consists of training and testing data with increasing complexity. Already available separated data for analysis.

6. The ground truth RULs are also available to analyze the results of predictions

7. Number 5T and 5V are more complex and bigger data sets which were used for a competition. The RULs were not available at first, but after the completion of the competition, the results were made public

8. These data sets are designed for fault degradation analyses and hence fit perfectly in the authors experimental setup

Figure 14: 90K turbofan degradation data-set types [54]

Datasets		#Fault Modes	#Conditions	#Train Units	#Test Units
Turbofan data from NASA repository	#1	1	1	100	100
	#2	1	6	260	259
	#3	2	1	100	100
	#4	2	6	249	248
PHM2008 Data Challenge	#5T	1	6	218	218
	#5V	1	6	218	435

Guidelines to process the data and build a fault degradation analysis is given in Figure A.5. This will be used as a reference for the process of developing a model.

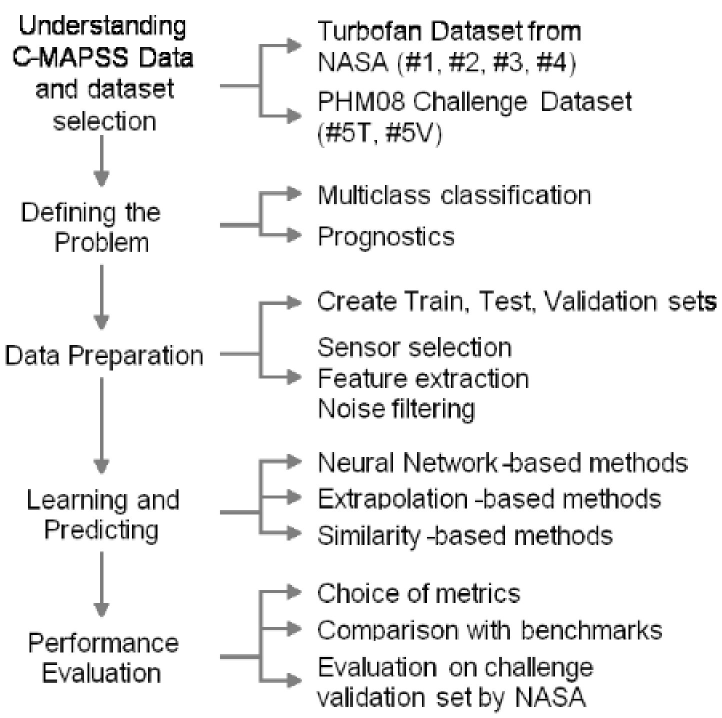

Figure 15: Guidelines to use C-MAPSS data sets [54]

Damage-propagation-model data

Referring to [58], the degradation models of the fault evolution have an exponential trend [32]. As physical-inspired data generation is concerned, the wear is calculated with Equation 9. It avoids the microlevel process but considers the macrolevel once.

$$w = Ae^{B(t)} \tag{9}$$

Furthermore, for the health index and the trajectories for flow and efficiency vary for different modes. The data has to be further addressed with different issues like wear to simulate the data in monotonic trends. These applications to be followed are initial wear, noise, data generation and health index calculation. After which, the data can be finally classified if the fault propagation trajectories cross the failure threshold. The health index calculates on how far the engine can be operated before stall or temperature limits are crossed. These failure modes would have to be compared with RUL and check scoring functions. Multiple models can be tested and checked for the best case possible.

3. Research aim

The goals of the proposed research based on the literature gap are discussed in this chapter. The research questions will be based on the research objectives of the thesis.

3.1. Research gap

As seen and discussed in research papers and articles in the previous sections, fault progression in aviation industry and specifically in predictive maintenance are dealt regularly. However, not much research was carried out in for improving the fault progression prognostics by improving the monotonic signals of the sensor data. The improvement techniques were usually used after the preprocessing of data with the help of decision tree models. Pre-processing of the data is carried out with feature engineering and methods like PCA to clean the trends. However, imposing monotonicity on the raw data to simplify and generate noise-free data for modeling is still not used.

3.2. Research objectives and scope

The objective is to check if imposing monotonicity on the data before inputting in the ML model would improve the predictions of RUL for fault progression. The scope is to analyze the proposed method and compare the results with existing preprocessing methods to see if the proposed method proves to have benchmarking outputs.

3.3. Research questions

Answering the following questions is of critical importance for achieving the goal of the thesis. SMART principle was used to generate these questions.

1. Is it possible to achieve monotonicity from non-monotonic signal dataset?
   (a) What is the monotonic metric value of the monotonic constraint signals? Is it entirely monotonic?
   (b) How does it compare with other pre-processing methods?
   (c) How much difference is noticed in the final values (end-of-life) of the monotonic signals of the component?
2. If the monotonic trends can be maintained, are they able to perform better prognostics? Which algorithm performs better and why?
   (a) How does it perform in terms of MAE, RMSE and R2?
   (b) What is the impact of the solution (in comparison)?
3. When considering the accuracy of a model, does the monotonically constrained model perform better than its counter part?
   (a) Which time-interval shows the least accuracy of monotonically constrained data? and why?
   (b) Does avoiding outliers help in accuracy of predictions?
   (c) Why does the monotonic constrained model perform better/worse?

4. Research Methodology

The project is planned to answer the research questions mentioned in the previous section with the help of the literature study. Modeling of the project and the approach for satisfying all research questions are discussed in this section.

4.1. Model approach

To first check if monotonicity in signals can be achieved, the ACD algorithm will be performed on the CMAPSS dataset. To check the results, a monotonic metric can be used to answer the first main research question. Other preprocessing methods should also be applied similarly and the results must be compared with the monotonically constrained dataset. Evaluation of monotonicity will give insights on the method of ACD. The dataset of CMAPSS will be splitted into 80-20 for training and testing results. The reason to do this split is to have the groundtruth values of RUL and the evaluation of the method can be observed through each sensor value.

It is also important to know how the newly constrained dataset would perform for prognostics. ML models which are time dependent can be used for the prediction of RUL. RUL estimation has to be established before estimating the predictions. The preprocessed datasets can be now trained and prognostics can be then performed for estimating RUL using different methods. The comparisons should reveal the observations for the proposed and already established methods. Evaluation metrics will be necessary for identifying the result.

5. Conclusion

The literature review provided an overview of the literature available and gave a structure to the methodology to be followed during the rest of the project. Unplanned maintenance has caused greater damage to the airline industry and data-based models is one of the solutions to this issue. By improving the prognostics, the window of probability distribution function of predictions will reduce and this report suggests multiple methods for it.

The non-monotonic signals present in fault progression were addressed by many authors to be unnecessary and therefore suggested to treat it with monotonicity for improving the prediction of RUL. The hypothesis presented in the literature review and the objective of the research are modeled in the approach. The objective of the research is to check if posing monotonicity on the non-monotonic data before inputting in the data into ML model would improve the predictions of RUL for fault progression.

Monotonic treatment technique ACD, is selected and will be used for experimenting with the CMAPSS dataset. The monotonic metric of Coble will be used for evaluating the range of monotinicity and the evaluation metrics will be used for establishing the performance of the models. After training of model, the model will be tested with a ML model and then planned to be validated with another ML model. These tests will be evaluated with a performance metric. The validity of the model will allow to test the model in the test data set. After evaluation of the new model, the other set of models are evaluated too and a comparison metric is defined. With the comparison metric, the pros and cons of the model can be evaluated and a conclusion can be made by answering the research questions. The work breakdown is defined in the following section.

6. Work-breakdown and Gantt chart

The work breakdown shall be carried out for the upcoming tasks of the thesis after the Kick-off meeting. The plan of execution for the given complexity of the problem is divided into 4 phases (as seen in Figure 16. At the end of each phase, a milestone is planned for keeping track of the work.

6.1. Phase 1 and 2

First two phases, namely "data preparation C-MAPSS" and "preparation for models" have the same ending period, the end of March. By the end of these phases, the author should have :

- Identified non-monotonic trends in the data
- Training, testing, and validation data must be separated and ready for modeling,

- A setup for recognizing the performance has to be modeled,
- The monotonic metric must be selected so that the best suit for estimating the monotonicity of a trend
- Selection of models from the case study need to studied and decided on which models and algorithms to use for comparison
- A verification of model and the algorithm needs to be setup

A delay of the maximum week is acceptable, but the modeling process has to start after that. Use the two meetings with supervisor that occurred in the phase.

6.2. Phase 3

The third phase represents the modeling period which begins from the end of March to the end of June (mid term).

- Algorithm for imposing monotonicity on trends, the model has to be tested on the data set
- Model testing for other ML algorithm
- The setup of the modeling would be designed as sprint. Each sprint would be for each model per week. At the end of a week, the model behavior and the results have to be reported.
- After validation, the knowledge has to be written in the report form and prepared for mid-term.

Phases 1, 2, and 3 must be completed by then. An extra week is given as a buffer week but included in the modeling phase itself. The midterm report has to be submitted before July begins.

6.3. Phase 4

Phase 4 lasts for just over 2 months. It begins in June and ends at the beginning of September.

- Include all suggestions and comments and submit the modeling and draft report
- Prepare for green light, and after that prepare for defense.

Green light meeting has to be done 1 month before the defense. Therefore, it has to be done before September begins.

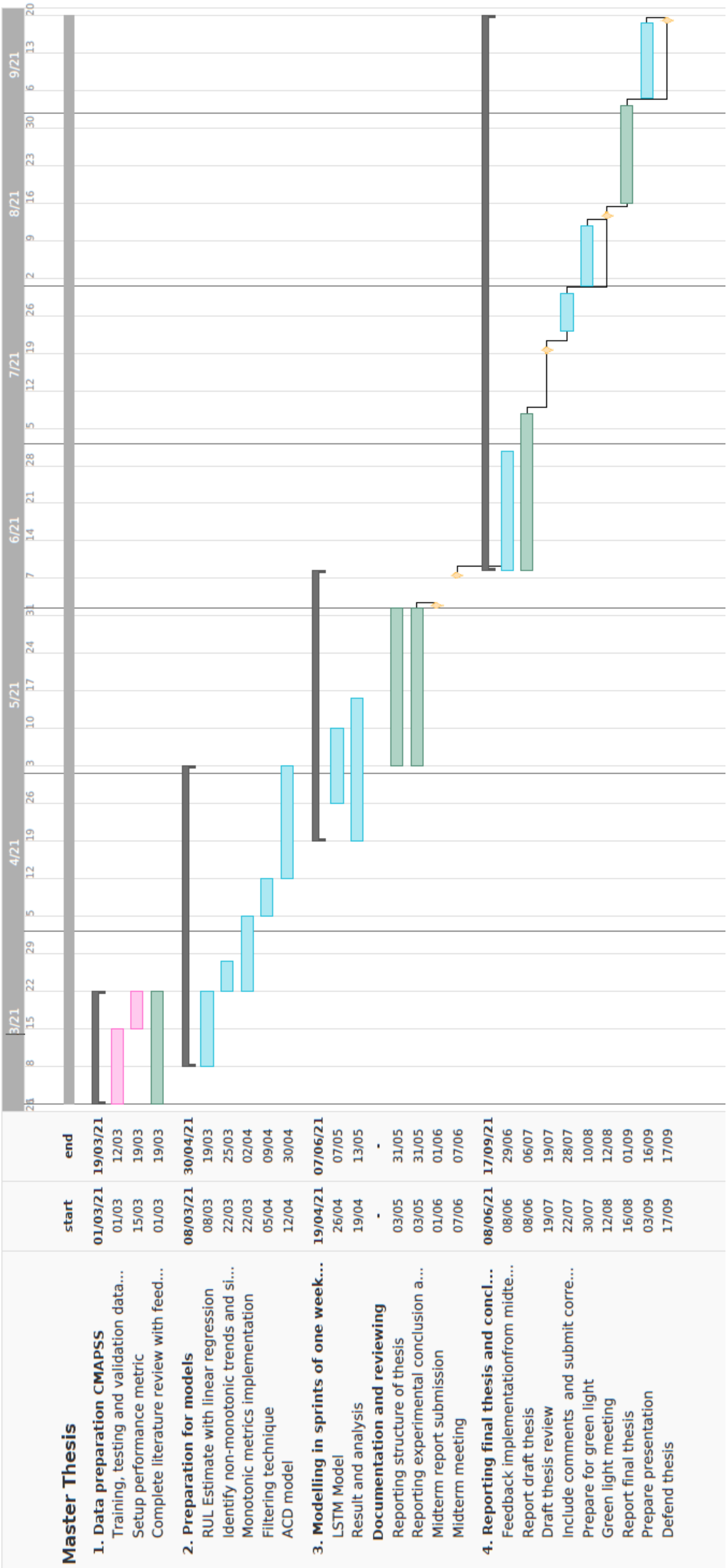

Figure 16: Project execution with the help of Gantt chart

III

Research Methodologies

1. Executive Summary

This research aims to investigate a state-of-the-art problem to treat the non-monotonic behavior of fault progression trends. Well-established algorithms and literature are researched for fault progression prognostics, however, not considerable attention has been given to monotonic constraints at a preprocessing stage. A non-monotonic trend carries complex information which has outliers and nonessential signal values. Such values can be refrained with monotonic constraints and only data which is insightful could be imported into the Machine learning (ML) algorithm. It is hypothesized that it will improve the accuracy of predictions. The goal of the project is to motivate the usage of monotonic constraints for improved and optimized predictive maintenance of a degrading component. Insights from literature studies and case studies will be explored to perform effective solutions on the test case of NASA CMAPSS dataset. The problem is presented as follows: Determining if the monotonic constrained method at a preprocessing step shall assist ML-algorithms to estimate the remaining useful life (RUL) of a component accurately, eventually optimizing maintenance activities and scheduling. For comparison, other preprocessing methods will be explored. Limitations and strengths will be studied and understood with the performance metrics.

2. Introduction

The scheduled regulatory aircraft maintenance are performed periodically and it serves the purpose of monitoring components of the aircraft. However, unexpected failures or unanticipated breakdowns of components cause disruption in operations, the cause of safety, and eventually affect the cost structure of the airline [34]. Preventive maintenance was performed for the longest time in the past and eventually the industry diverted towards predictive maintenance for its cost benefits and effective operations [15]. Operations of maintenance of aircrafts are very complex and require attention to detail [26]. Constant research and experiments are carried out to perfect the prognostics and perform predictive maintenance robustly and effectively. This research aims to add further to improve the predictive maintenance for degrading components of an aircraft.

The prognostics and health management (PHM) have multiple approaches to perform prognostics and schedule accurate maintenance. In this research, the data-driven approach is opted for its cost-effective benefits, growing interest, and popularity in the industry. The materials and components that degrade overtime theoretically have a logarithmic behavior (as seen in Figure 1) in their sensor data. Due to the noise and disturbances processed by the sensors of the component, monotonic trends are not obtained. These uncertainties cause failure in prediction as the ML-model possess unwanted extra information. Therefore, through this project plan, the author proposes the steps planned to improve the uncertainty by treating the non-monotonic signals with monotonic signals and validating better performance.

The literature review in Section 2 explores four researched topics: non-monotonic and monotonic signals, monotonic improvement techniques, preprocessing steps in ML, and evaluating the monotonic signals and their performances. The conceptual research model is presented with the research questions and objectives in Section 3. The methodology and experimental setup are discussed in Section 4 and Section 5, respectively. Finally, the expected outcome of the research is in Section 6 and the work break-down is explained in a Gantt chart in Section 7.

3. Literature Review

The scope of the research proposal is discussed before hand. The research is focused on identifying the benefits of treating non-monotonic signals to monotonic signals for fault progression data. Key concepts of the research include the importance of monotonicity in data, identifying the monotonicity of signals, and monotonic improvement techniques and preprocessing. Evaluation and performance techniques are discussed towards the end of the section. These concepts will be reviewed and discussed in this section.

3.1. Prognostics and health management

Prognostics ensure detecting and informing system faults and failures for maintenance. Along with fault detection and fault isolation, PHM detects fault prediction on selected components, fault filtering and reporting, predicting RUL, health management, and recommended actions to pilot when necessary [27]. PHM process has different stages and features; namely, raw data from sensors, diagnostics, prognostics, and health management. These are modeled for determining the health stage of a component. The PHM process can be modeled

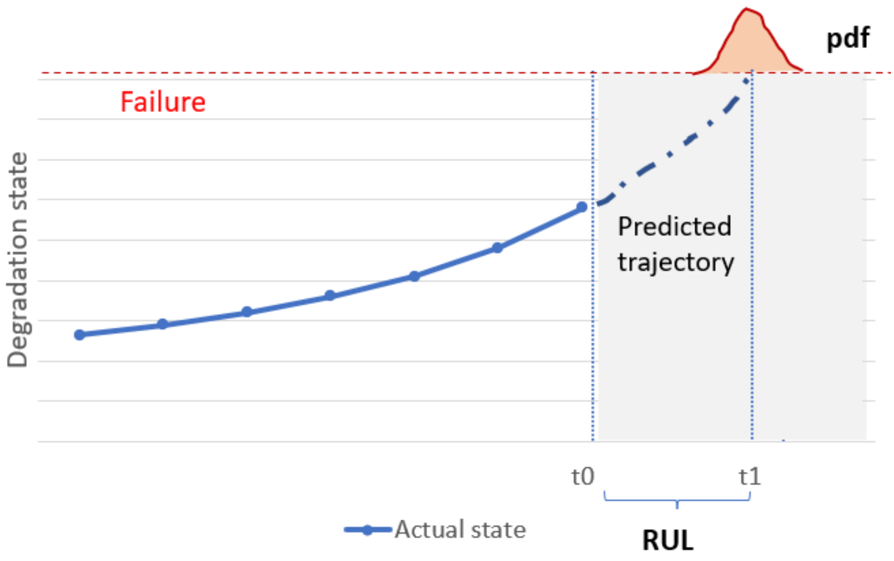

Figure 1: Degradation state of a component and its estimated prediction of RUL after its actual state

for two different types of maintenance. A model from Sandia National laboratories (SNL) [56] explains the difference of evidence and consequence maintenance scenarios [?]. The evidence establishes the feature extraction, trend detection, and estimation of RUL while the consequence analyzes the end result of maintenance actions. In the research, the evidence scenario is going to be explored, and specific cases of RUL estimation. The reason is to explore the component's life and determine the best time for scheduling maintenance to avoid accidents and prolong its life.

RULs provide decision makers with information about the component health life. The estimation of RUL helps scheduling the maintenance before time. The algorithms of prognostics predicting RUL are classified as degradation-based prognostics [16]. The degradation-based modeling of a system or component is defined as the length from the current time to the end of its useful life and it can be used to characterize the system's current health status [13]. The RUL value can be identified in the Figure 1 from duration of $t0$ to $t1$.

RULs can be estimated with two approaches in prognostics, model-based approach and data-driven approach [13]. Examples of model-based approaches are physical failure models, stochastic filtering models, and statistical models. Data-driven approaches use past data to predict the future and their examples are ML models and graphical models. Both approaches are used in PHM. The advantage of model-based approaches is the ability to understand the physical fundamentals of a monitored system and therefore have higher accuracy in prediction. Whereas the source of understanding of data-driven approaches is through the measured input/output data from sensors. The data-driven approaches are less accurate in comparison to model-based approaches as the data of the components are complex, noisy, and full of uncertainty [13]. On-the-other-hand, model-based approaches require experienced engineers who have an understanding of each component's physical state and can operate to extend their life cycle. However, this process requires specific tools, machinery, and experience players to operate and therefore is costly. Data-driven approaches have seen immense growth in the development of methods to improve the accuracy of predictions and are inexpensive in comparison to model-based approaches. This research should further add to the development and existing body of knowledge of data-driven approaches.

3.2. Non-monotonic and monotonic signals

Signals from the sensor data of the components describe the health state and the conditions it has gone through. Multiple cycles of the sensor data provide patterns for the ML-algorithm to learn and estimate the RUL of a component. A component's life degrades over time and reaches it's end of life. The health of the component should linearly decrease to its end of life theoretically. However, many discrepancies in data cause the signal to be nonlinear and the degradation trend is not monotonic. The non-monotonicity of the

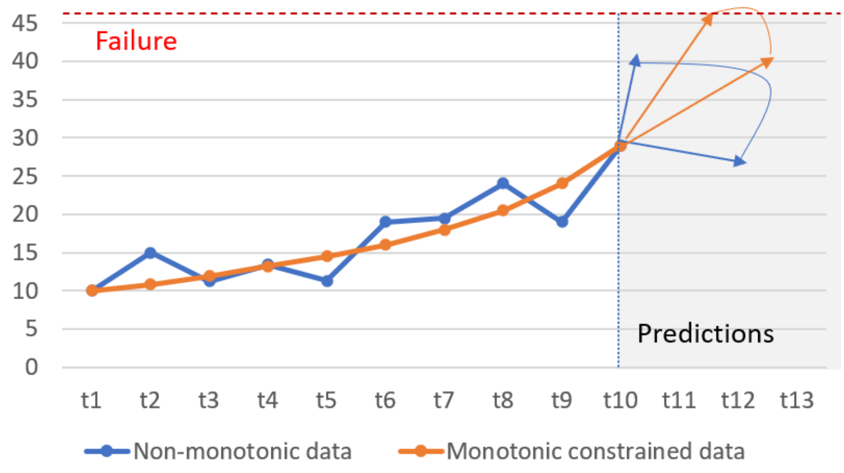

Figure 2: Hypothesis of prediction window of monotonic signals vs non-monotonic signals

signal causes difficulty in identifying patterns for an ML algorithm, especially towards end of the component's life. This data could deviate from the hypothesized results and is thus unreliable for prediction. Hence, causes wrong maintenance scheduling. A visual example is shown in Figure 2 where the predictions from monotonic constrained data have a smaller size of the hypothesis and are thus more powerful for degrading components.

In principle, monotonic signals do not change their directions, the signal will maintain one direction. On the contrary, non-monotonic signals will change direction and values as more time cycles are added and invalidate the previous conclusions. The signals of a non-monotonic dataset carry complex information which are not insightful for prediction. The data are scattered and do not necessarily follow a pattern for the ML-model to train and replicate for fault progression.

Irregularity and uncertainty are the main causes of failing PHM models [48]. Imperfections in the predictability of prognostic models can be caused from three different sources, according to Baraldi [4]; Randomness related to future degradation, modeling errors and inaccuracies in degradation data. Researchers of [38] emphasizes how monotonicity in data can avoid the exhaustive search without sacrificing optimality. It is also recommended by [16] and [1] to quantify and consider monotonicity in the systematic construction of PHM models. Feelders [20], claims that models trained on monotonic datasets often have better predictive performance than models trained on original data. Monotonic datasets could be created by generating artificial data or by relabeling of real data [41][12][29].

Monotonicity as a property states that an increase in input cannot result in a decrease in the related output [5]. So, adding a monotonic constraint to a model would reconstruct the trends of the model to guarantee a monotone relation between explanatory variables and dependent variables. Monotonicity poses a monotonic trend where the condition is fulfilled if and only if the trend is either entirely non-increasing or entirely non-decreasing. In other words, if the signal is monotonically increasing or decreasing with time, corresponding to an improving or deteriorating system, there is supposed to be a monotone trend, otherwise the trend is non-monotone [30]. This behavior of the trend would be eventually useful for prognostics as the trend would reduce the uncertainty and can deduce the actual life cycle of the component for the algorithm to learn and predict. It was also proven in [62] paper that monotone prediction outperforms the standard counterparts due to successfully avoiding overfitting.

On-the-other hand, the paper by Ben-David mentions that adding monotonic constraints to ordinal regressors can reduce their accuracy. In this case, the author of [5] says that the explanatory variables in the data sets used for the experiments were not in a monotone relationship with the labels. Therefore, making use of monotone models inappropriately results in a poor performance. The author further says that the benefits of monotone or partially monotone data models can be fully beneficial when one can be sure of the relations present in the data.

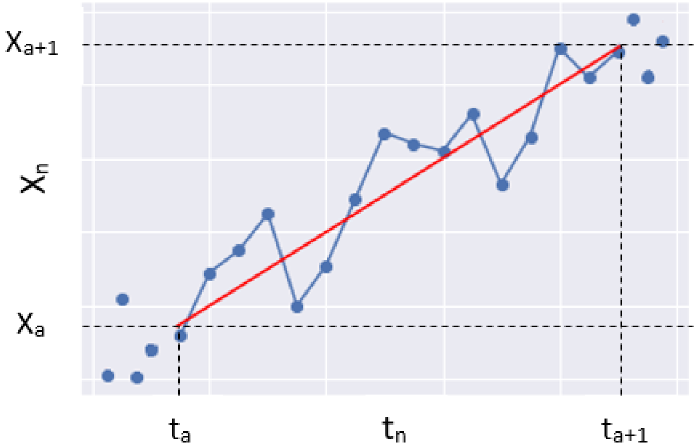

Figure 3: Application of Average conditional displacement(ACD) on an interval of non-monotonic signal [48]

3.3. Monotonic improvement techniques

Techniques were researched in depth for choosing the apt method for the research. They are discussed in this section.

Enforcing monotonicity of decision models for data mining: Purpose of data mining is to derive the right information from available datasets to propel decision making. Credit loan approvals, risk analysis need the models to be monotone in relation to decision variables. The authors of [67] propose a method to enforce monotonicity and clean up non-monotonic data by maintaining monotonic relationship between dependent variables and explanatory variables for domain knowledge. They relabel the non-monotonic dependent variables into monotonic one with the degree of monotonicity of data and an algorithm to clean non-monotone data sets. The performance achieved was better and proved its capability in economic decision making.

Non-parametric time series are used to develop and extract features representing the nominal behavior of the monitored component and derive smoother trends to represent critical components of health evolution over time [46]. Empirical mode decomposition algorithm is applied and ridge regression to extract the trend for RULs. The developer of the model was successful in demonstrating a smooth accelerated degradation dataset.

Average conditional Displacement (ACD) introduced by Vamos [65]. The algorithm is based on a signal value interval, but it works fairly well for estimating the monotonic trends of a stationary noise-filled time series data [48]. The algorithm is used for removing monotone trend from non-monotonic data and reveals one of the possible monotonic version of the non-monotonic trend. The advantage to use this algorithm is two-folded. First, this algorithm does not require any initial subjective assumptions and approximates monotonic trends as a piecewise linear curve by dividing into sub-intervals of signal intervals [65]. Secondly, it is an automated algorithm which is comparable with known methods like moving average and polynomial fitting [48].

The algorithm considers small intervals randomly placed across a trend. The extreme ends of the intervals are considered and an average line is drawn through the points. This is illustrated in the Figure 3. The figure describes a small interval of a signal from a sensor. The blue dots represent the original sensor value and the red thick line represents the monotonic signal after application of ACD approximation for the interval. The average sample of the slope is estimated with Equation 1[66]. Where, x_n represents the pieces of interval, N_a is the number of x_n values of the interval and g is the slope. The iterations of the intervals are carried out throughout the trend for a smooth and monotonic curve. The author of [48] says the ratio between the noise fluctuation and the amplitudes of trend variation is the major contributor to the accuracy of the ACD algorithm. ACD has the potential benefits to treat non-monotonic signals and stationary noise in PHM applications.

$$\hat{g} = \frac{1}{2N_a}\left(\sum_{x_n \epsilon I_a} \delta x_n + \sum_{x_{n+1} \epsilon I_a} \delta x_n\right) \tag{1}$$

Monotonic classification and regression model approaches have been considered in the literature of [53] and [52]. Also, monotonic neural networks in [24] and hybridization. These models were successful in predicting accurate prognostics compared to their counterparts with non-monotonic models. The question is raised from the techniques mentioned before, if one has the dataset constrained to monotonicity already and applies a regular regression model to it, should it perform better in comparison to the dataset that was not constrained? This is the hypothesis for the research. In theory, the simplification and filtration beforehand would reduce the computational time and yield the results of predictions in simple steps. Having a clear trend of the time for a component to degrade would be easier to learn than an uncertain one. Therefore, a monotonic constraint at a preprocessing step is advised through this research. A comparison with other preprocessing methods should be made too.

3.4. Preprocessing

Data cleaning has been of interest since the growing size of the database and the requirement for turning the data into useful knowledge. Preprocessing is a type of data cleaning in prognostics, which has been important for reducing randomness in data and importing the right information into the algorithm. This procedure reduces the computational time and increases accuracy in prognostics. There are different methods of preprocessing. For example, feature engineering is a process where unclassified and unlabeled data are clustered into a group with similar trends or examples. It results in identifying correlations and dependencies existing between their features [7]. Similarly, for analyzing features, Principal component analysis (PCA) is available. The goal of PCA is to analyze the covariance and reduce the complexity of data. In [48], the PCA was used for smoothing acoustic emission signals by eliminating highly correlated variables. Smoothing is an extracting technique applied to time series to extract variations between time steps [7].

Preprocessing step is also responsible to reduce the noise in the signal trends. To identify and learn the behavior of a signal during the training of the model, the high noise in the signal would be incorporated and studied. However, for fault progression and identifying the RUL of a component, the noise in the data causes uncertainty. It does not add value for identifying the end of life. It merely increases the computational steps and the hypothesis window of prediction (as seen in Figure 2). There are methods available which help to reduce the noise, such as rolling mean or moving average. Rolling mean operations results in cleaner and understandable trends for the ML model to learn and perform prognostics accurately. However, these processed trends still have distortion and noise [7]. Moving average develops a new series with average values of raw data along the time series. It assumes the time series to be stationary while computing and the signal does not have monotonic behavior or seasonality. The moving average depends on the window width defined by the raw observations and calculates the average per window width [7]. These preprocessing steps will be compared with the monotonic constrained model.

Adding a monotonic constraint during a preprocessing step to a fault progression data type is hypothesized to result in a smoother and monotonic dataset. The updated dataset of the components state replaces a healthy with an unhealthy state within time, the features of the component show a degrading trend in the data. In other words, changing monotonously in the decreasing propagation. Any regressor model can be used to train the model.

3.5. Evaluation

Performance metrics like Root mean square error (RMSE), Mean absolute error (MAE) can determine the prediction errors and how far are they off. However, to measure the monotonicity of a signal, the following metrics were researched.

Coble [16]: showcases the Local Gradient-Based method as the "The average difference of the fraction of positive and negative derivatives of the time series of a feature over time". The monotonic metric is calculated with Equation 2. Where G^+ and G^- represent vector positive and negative derivatives respectively (seen in Equation 3). Δy represents the difference between two consecutive points on the signal.

$$\mathrm{MM}_1 = \mathrm{mean}\left(\left|\frac{\#\left[G^+\right]}{N-1} - \frac{\#\left[G^-\right]}{N-1}\right|\right) \tag{2}$$

where,

$$\begin{aligned} G_i^+ &= \frac{\Delta y_i^+}{\Delta t_L} = \frac{y_i - y_{i-1}}{\Delta t} \mid \left(y_i - y_{i-1}\right) > 0 \\ G_i^- &= \frac{\Delta y_i^-}{\Delta t_i} = \frac{y_i - y_{i-1}}{\Delta t} \mid \left(y_i - y_{i-1}\right) < 0 \end{aligned} \tag{3}$$

The same formula can be rewritten as Equation 4 [16]. The differences of adjacent points of a signal are compared for identifying monotonicity. N_j is representing the total points, $x_j(k)$ represents a point of the signal and $x_j(k+1)$ is the subsequent point. Each step of the signal is compared and an average score is determined. For each trend, the monotonicity adjusts from a value of zero to one. Zero being the least monotonic and one with the highest monotonicity. The formula would be used for checking the monotonicity of the signal. It will also be useful for comparing different preprocessing datasets for their monotonicity check.

$$\text{Monotonicity} = \frac{1}{M}\sum_{j=1}^{M}\left|\sum_{k=1}^{N_j-1}\frac{\operatorname{sgn}\left(x_j(k+1) - x_j(k)\right)}{N_j - 1}\right| \tag{4}$$

Verification based testing: The technique used in [60], approximates the black-box model by a white box model. SMT solving techniques are used for computing the monotonicity metric of white-box model. "The term black-box is mainly used for labeling all those machine learning models that are (from a mathematical point of view) very hard to explain and to be understood by experts in practical domains."[39][55] "The terms white-box, understandable model, and explainable artificial intelligence (XAI) are used for labeling all those machine learning models, providing results associated to their models that are easy to understand by experts in the application domain. Usually, these models provide a good trade-off between accuracy and explainability."[39][55]

4. Research Question, Aim/Objectives and Sub-goals

This section discusses the goals of the thesis project. The goals are explained through implementing research questions and objectives.

4.1. Research Question(s)

1. Is it possible to achieve monotonicity from non-monotonic signal dataset?
    (a) What is the monotonic metric value of the monotonic constraint signals? Is it entirely monotonic?
    (b) How does it compare with other pre-processing methods?
    (c) How much difference is noticed in the final values (end-of-life) of the monotonic trend of the component?
2. If the monotonic trends can be maintained, are they able to perform better prognostics? Which algorithm performs better and why?
    (a) How does it perform in terms of MAE, RMSE and R2?
    (b) What is the impact of the solution (in comparison)?
3. When considering the accuracy of a model, does the monotonically constrained model perform better than its counter part?
    (a) Which time-interval shows the least accuracy of monotonically constrained data? and why?
    (b) Does avoiding outliers help in accuracy of predictions?
    (c) Why does the monotonic constrained model perform better/worse?

4.2. Research Objective

"To investigate if applying monotonic constraints to non-monotonic trends of a fault progression dataset can improve prediction accuracy of the ML-model for predicting RULs"

The objective of the research focuses on improving maintenance scheduling and operations process with optimized ML-model. The objective is achieved by insights from literature, experimentation of the proposed models and evaluation of the method. It is firstly crucial to check if applying an algorithm in preprocessing step can produce monotonic constrained datasets. Thereafter, the performance of the model has to be evaluated. It can be achieved with performance metrics and comparison with competitive methods.

5. Theoretical Content/Methodology

There are multiple methods available to test and figure the research questions. In this research, the method will be to focus on using a monotonic constrained algorithm in the preprocessing step to strategically reduce the unwanted data and improve the prediction further for the degrading component.

It is important to understand that this thesis focuses on proving monotonicity in data, specifically fault progression data, can improve overall accuracy of predictions. Monotonic data models need to be compared with non-monotonic data models, as well as other types of preprocessing methods for in-depth evaluation and comparison. The original dataset shall be split into training and testing datasets. Multiple versions of this training dataset will be generated by applying the different methods proposed in the literature. First will be the original dataset untouched. The second will be the monotonically constrained dataset with the ACD algorithm, then the one with moving average and the last one with rolling mean. Each type of trained dataset will be subjected to the same random forest regressor(ML-algorithm). The testing dataset will remain common and unchanged to compare the outcomes and evaluate results.

RUL formulations need to be programmed for prediction. The RUL estimation method will remain constant for each data type. Lastly, it is important to check different ML methods to analyze the best performance. Therefore, the case of CNN-LSTM algorithm will be programmed to validate the results of the random forest regressor. Optimized solution will directly impact the objective of the research, and the results will validate the method to be useful for maintenance methods or not.

6. Experimental Set-up

The experiment is designed for estimating if monotonically constrained data can successfully improve prognostics and to analyze their behavior. The dataset of NASA CMAPSS of turbo-fan engine will be split into training and testing datasets with the ratio 8:2. The reason for splitting the training and testing datasets is to have the actual values of the test signals to analyze and compare the results. The implementation of ML-algorithm will be carried out in Python. A MATLAB program is required for ACD. An input of the training dataset will be made to the constraint method and the resulting monotonically constraint dataset will be the output. This data will be compared with the original non-monotonic dataset and other performed preprocessed methods.

For analyzing the results, first a monotonic metric is selected to identify if a monotonic behavior can be obtained from a non-monotonic signal. Secondly, the analysis will be carried out on performance metrics and scoring functions. Over-estimation and underestimation can be understood based on the results of the scoring values. Furthermore, alpha-lambda curves will be used to identify the accuracy of the models.

7. Results, Outcome and Relevance

If altering data to monotonically constrained signals can perform optimized results that would positively answer all three research questions. It would validate the methodology and will be a step forward for PHM. The results of the model should be accessed further if performed and must be recorded for further analysis. It is also important to know "why" for the outcomes of the model. Therefore, a deeper analysis on the performance will guide the answer for the results of the model. The performance for each research question would yield important insights for key indicators. The project is limited for fault progression trends.

It is paramount to verify and validate the optimized model, and therefore two ML-algorithms will be tested on the same dataset to validate the model. Lastly, the system tests for the model and the complete models have to be verified.

8. Project Planning and Gantt Chart

The project planned during the literature review phase is showcased in Appendix A. From preparing the model and datasets to the reporting of the final paper has been planned and shall be carried out for maximum efficiency. The meetings with the supervisor until now have been carried out online due to COVID. The meetings are carried out biweekly and should be continued to do so during the rest of the project. Except for a sprint phase during the midterm which shall have meetings weekly with the supervisor.

9. Conclusions

The project plan for the treatment of non-monotonic signals in fault progression of a turbo-fan engine combines the thorough literature review, conceptual research design, and technical research design. The aim of the proposal is to add value to data-driven approaches of predictive maintenance by improving the data processed from the sensors. The goal derived from this project plan is to determine if the monotonic constrained method at the preprocessing step shall improve the predictions of ML algorithm to estimate RUL of the fault progression dataset.

The insights gained from the literature review showcased considerable support for the treatment of non-monotonic signals. Researchers described the irregularity and uncertainty of data causes failure in the predictions of RUL values. Increasing the quality of data improves the prognostics process and the models can be trained effectively. Multiple monotonic improvement techniques were proposed in the literature which improves quality of data. Introducing monotonicity of data in data mining, monotonic classification models, monotonic neural networks, and enforcing monotonicity in the preprocessing step of prognostics were a few addressed in the review. For this project, it is decided to use the ACD model in the preprocessing step to treat non-monotonic signals and confirm the benefits of treating non-monotonic data by implementing them in ML-models.

The method chosen is to optimize the process for modeling a fault progression ML model. The analysis of the results, limitations and evaluations will be carried out with performance metrics and additional monotonic metrics. The NASA CMAPSS dataset of a turbofan engine will be used for experimentation. The model will be coded in MATLAB and Python. The model will be tested with 2 ML algorithms. Finally, the proposed project is planned into 4 steps for completion in the next 7 months.

Having the research performed, a notable contribution will be added to the research catalog and it shall fill up the answer to the question which is still empty. No research has been carried out on CMAPSS dataset with ACD algorithm in the preprocessing step for an ML-algorithm.

10. Gantt Chart

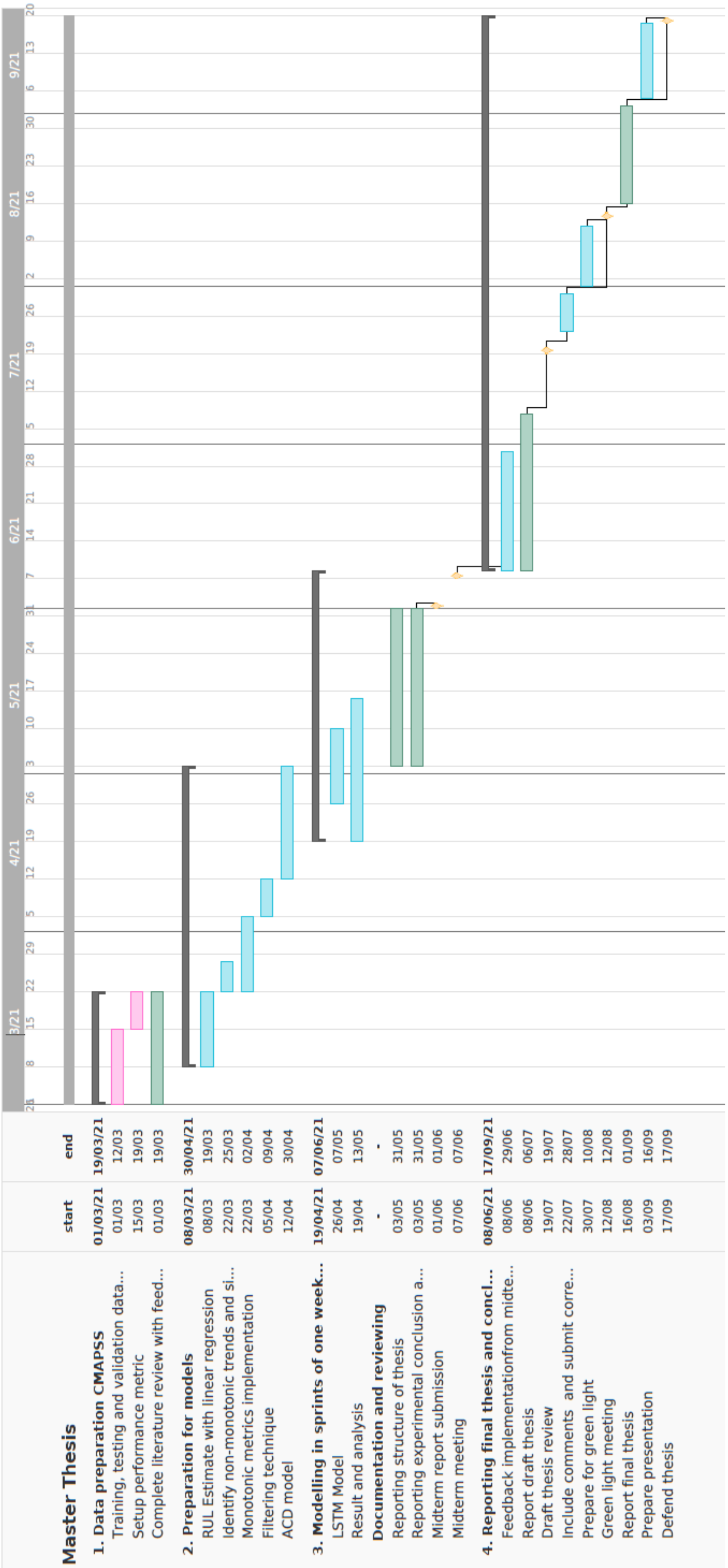

Figure 4: Project execution described with gantt chart

Bibliography

[1] J.I. Aizpurua and VM Catterson. Methodology for Design of Prognostic Systems. 2015.

[2] Giduthuri S. Babu, P. Zhao, and Xiao Li. Deep convolutional neural network based regression approach for estimation of remaining useful life. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9642, pages 214–228. Springer Verlag, 2016. ISBN 9783319320243. doi: 10.1007/978-3-319-32025-0_14.

[3] Mohua Banerjee, Sushmita Mitra, and Sankar K. Pal. Rough fuzzy MLP: Knowledge encoding and classification. *IEEE Transactions on Neural Networks*, 9(6):1203–1216, 1998. ISSN 10459227. doi: 10.1109/72.728363.

[4] P. Baraldi and E. Zio F. Mangili. Investigation of uncertainty treatment capability of model-based and data-driven prognostic methods using simulated data. 112:94108, 2013.

[5] T. Bartoji. Testing Monotonicity of Variables. 2010.

[6] Arie Ben-David. Monotonicity maintenance in information-theoretic machine learning algorithms. *Machine Learning*, 19(1):29–43, apr 1995. ISSN 0885-6125. doi: 10.1007/bf00994659.

[7] J. Brownlee. https://machinelearningmastery.com/moving-average-smoothing-for-time-series-forecasting- 2016.

[8] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering*. Cambridge University Press, jan 2019. doi: 10.1017/9781108380690.

[9] Steven L Brunton, J Nathan Kutz, Krithika Manohar, Aleksandr Y Aravkin, Kristi Morgansen, Jennifer Klemisch, Nicholas Goebel, James Buttrick, Jeffrey Poskin, Agnes Blom-Schieber, Thomas Hogan, and Darren Mcdonald. Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning. Technical report, 2008.

[10] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine Learning for Fluid Mechanics. *Annual Review of Fluid Mechanics*, 52(1):477–508, jan 2020. ISSN 0066-4189. doi: 10.1146/annurev-fluid-010719-060214.

[11] F. Camci, K. Medjaher, N. Zerhouni, and P. Nectoux. Feature evaluation for effective bearing prognostics. *Quality and Reliability Engineering International*, 29(4):477–486, jun 2013. ISSN 07488017. doi: 10.1002/qre.1396.

[12] J. Cano, J. Luengo, and S. Garca. Label Noise Filtering Techniques to Improve Monotonic Classification. 2018.

[13] Xiongzi Chen, Jinsong Yu, Diyin Tang, and Yingxun Wang. Remaining useful life prognostic estimation for aircraft subsystems or components: A review. In *Proceedings - IEEE 2011 10th International Conference on Electronic Measurement and Instruments, ICEMI 2011*, volume 2, pages 94–98, 2011. ISBN 9781424481590. doi: 10.1109/ICEMI.2011.6037773.

[14] Z. S. Chen, Y. M. Yang, and Zheng Hu. A technical framework and roadmap of embedded diagnostics and prognostics for complex mechanical systems in prognostics and health management Ssystems. *IEEE Transactions on Reliability*, 61(2):314–322, 2012. ISSN 00189529. doi: 10.1109/TR.2012.2196171.

[15] L. Ciocoiu, C. Siemieniuch, and E. Hubbard. From preventative to predictive maintenance: The organisational challenge. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 231:1174–1185, 11 2017. ISSN 20413017. doi: 10.1177/0954409717701785.

[16] J. Coble and Baalis. Trace: Tennessee Research and Creative Exchange Merging Data Sources to Predict Remaining Useful Life-An Automated Method to Identify Prognostic Parameters Recommended Citation. 2010. URL https://trace.tennessee.edu/utk{_}graddiss/683.

[17] Datadive. Monotonicity constraints in machine learning | Diving into data. URL https://blog.datadive.net/monotonicity-constraints-in-machine-learning/.

[18] EXSYN. Predictive Maintenance | EXSYN. URL https://www.exsyn.com/blog/big-data-in-aviation-predictive-maintenance.

[19] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. Knowledge Discovery and Data Mining: Towards a Unifying Framework. Technical report, 1996.

[20] A. Feelders. Monotone relabeling in ordinal classification. pages 803–808, 2010.

[21] Dean K Frederick, Jonathan A Decastro, and Jonathan S Litt. User ' s Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) October 2007. 2019.

[22] Alex A. Freitas. A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery. pages 819–845. Springer, Berlin, Heidelberg, 2003. doi: 10.1007/978-3-642-18965-4_33.

[23] Rafael Gouriveau, Kamal Medjaher, and Noureddine Zerhouni. *From Prognostics and Health Systems Management to Predictive Maintenance 1: Monitoring and Prognostics*, volume 4. wiley, oct 2016. ISBN 9781119371052. doi: 10.1002/9781119371052.

[24] M. Velikova H. Daniels. Monotone and partially monotone neural networks. page 906917, 2010.

[25] Frank E. Harrell, Kerry L. Lee, and Daniel B. Mark. Multivariable prognostic models: Issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in Medicine*, 15(4):361–387, feb 1996. ISSN 02776715. doi: 10.1002/(SICI)1097-0258(19960229)15:4<361::AID-SIM168>3.0.CO;2-4.

[26] I.A. Herrera, A. O. Nordskag, G. Myhre, and K. Halvorsen. Aviation safety and maintenance under major organizational changes, investigating non-existing accidents. *Accident Analysis Prevention*, 41: 1155–1163, 11 2009. ISSN 0001-4575. doi: 10.1016/J.AAP.2008.06.007.

[27] A. Hess and L. Fila. The joint strike fighter (jsf) phm concept: Potential impact on aging aircraft problems. *IEEE Aerospace Conference Proceedings*, 6:3021–3022, 2002. ISSN 14333015. doi: 10.1109/AERO.2002.1036144.

[28] Qing Hua Hu, Mao Zu Guo, Da Ren Yu, and Jin Fu Liu. Information entropy for ordinal classification. *Science in China, Series F: Information Sciences*, 53(6):1188–1200, jun 2010. ISSN 10092757. doi: 10.1007/s11432-010-3117-7.

[29] R. Potharst I. Milstein, A. Ben-David. Generating noisy monotone ordinaldatasets. *Artificial Intelligence Research*, 3:30–37, 2014.

[30] J. T. Kvaloy and B. Lindqvist and H. Malmedal. A statistical test for monotonic and non-monotonic trend in repairable systems. *Proc European Conference on Safety and Reliability ESREL 2001 Torino 2002*, (May 2014):1563–1570, 2002.

[31] Kamran Javed. based on extreme learning machine and fuzzy clustering . Thèse de Doctorat. 2014.

[32] N. Eklund K. Goebel, H. Qiu and W. Yan. Modeling Propagation of Gas Path Damage. *IEEE Aerospace Conference*, 2007.

[33] Kwang Gi Kim. *Deep learning*. 2016.

[34] B. Kitchenham, S. Pfleeger, B. McColl, and S. Eagan. An empirical study of maintenance and development estimation accuracy. *Journal of Systems and Software*, 64:57–77, 10 2002. ISSN 01641212. doi: 10.1016/S0164-1212(02)00021-3.

[35] Teuvo Kohonen, Samuel Kaski, Krista Lagus, Jarkko Salojärvi, Jukka Honkela, Vesa Paatero, and Antti Saarela. Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11 (3):574–585, may 2000. ISSN 10459227. doi: 10.1109/72.846729.

[36] Panagiotis Korvesis. Machine Learning for Predictive Maintenance in Aviation. 2019.

[37] Sumit Kundu, Michael Fowler, Leonardo C. Simon, and Rami Abouatallah. Reversible and irreversible degradation in fuel cells during Open Circuit Voltage durability testing. *Journal of Power Sources*, 182(1): 254–258, jul 2008. ISSN 03787753. doi: 10.1016/j.jpowsour.2008.04.009.

[38] H. Liu, H. Motoda, and M. Dash. A monotonic measure for optimal feature selection. `https://link-springer-com.tudelft.idm.oclc.org/content/pdf/10.1007/BFb0026678.pdf`. (Accessed on 10/11/2021).

[39] Octavio Loyola-Gonzalez. Black-box vs. White-Box: Understanding their advantages and weaknesses from a practical point of view, 2019.

[40] Vyacheslav Lyubchich, Yulia R. Gel, and Abdel El-Shaarawi. On detecting non-monotonic trends in environmental time series: A fusion of local regression and bootstrap. *Environmetrics*, 24(4):209–226, jun 2013. ISSN 11804009. doi: 10.1002/env.2212.

[41] H. De Meyer M. Rademaker, B. De Baets. Optimal monotone relabelling of partially non-monotone ordinal data. *Optimization Methods and Software*, pages 17–31, 2012.

[42] Sushmita Mitra and Sankar K. Pal. Fuzzy Multi-Layer Perceptron, Inferencing and Rule Generation. *IEEE Transactions on Neural Networks*, 6(1):51–63, 1995. ISSN 19410093. doi: 10.1109/72.363450.

[43] Sushmita Mitra, Sankar K. Pal, and Pabitra Mitra. Data mining in soft computing framework: A survey. *IEEE Transactions on Neural Networks*, 13(1):3–14, jan 2002. ISSN 10459227. doi: 10.1109/72.977258.

[44] R. Keith Mobley. An Introduction to Predictive Maintenance , 2002.

[45] Davood Mosaddar and Amir Abbas Shojaie. A data mining model to identify inefficient maintenance activities. *International Journal of Systems Assurance Engineering and Management*, 4(2):182–192, jun 2013. ISSN 09756809. doi: 10.1007/s13198-013-0148-7.

[46] Ahmed Mosallam, Kamal Medjaher, and Noureddine Zerhouni. Nonparametric time series modelling for industrial prognostics and health management. *International Journal of Advanced Manufacturing Technology*, 69(5-8):1685–1699, 2013. ISSN 02683768. doi: 10.1007/s00170-013-5065-z.

[47] Javaid Nabi. Machine Learning Fundamentals. Basic theory underlying the field of Machine learning, 2018. URL `https://towardsdatascience.com/machine-learning-basics-part-1-a36d38c7916`.

[48] Seyed A. Niknam, John Kobza, and J. Wesley Hines. Techniques of trend analysis in degradation-based prognostics. *International Journal of Advanced Manufacturing Technology*, 88(9-12):2429–2441, 2017. ISSN 14333015. doi: 10.1007/s00170-016-8909-5.

[49] Seyed Ahmad Niknam. Prognostic-based Life Extension Methodology with Application to Power Generation Systems. pages 70–90, 2014.

[50] Agusmian Partogi Ompusunggu. On Improving the Monotonicity-Based Evaluation Method For Selecting Features / Health Indicators for Prognostics. pages 3–7, 2020.

[51] S. L. Kelly P. S. Laura, E. C. James and B. D. Adele. Algorithm Development for Prognostics and Health Management. Oct 2003.

[52] R. Potharst and A. J. Feelders. Classification trees for problems with monotonicity constraints. *ACM SIGKDD Explorations Newsletter*, 4(1):1–10, 2002. ISSN 1931-0145. doi: 10.1145/568574.568577.

[53] J. Bioch R. Potharst. Decision trees for ordinal classification. pages 97–111, 2000.

[54] Emmanuel Ramasso, Abhinav Saxena, Emmanuel Ramasso, Abhinav Saxena, Performance Benchmarking, and Analysis Prognostic Meth. Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets. 2016.

[55] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, may 2019.

[56] E. C. James S. L. Kelly P. S. Laura and B. D. Adele. Algorithm development for prognostics and health management. 2003.

[57] Abhinav Saxena, Jose Celaya, Edward Balaban, Kai Goebel, Bhaskar Saha, Sankalita Saha, and Mark Schwabacher. Metrics for evaluating performance of prognostic techniques. In *2008 International Conference on Prognostics and Health Management, PHM 2008*, pages 1–17. IEEE, oct 2008. ISBN 9781424419357. doi: 10.1109/PHM.2008.4711436.

[58] Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management, PHM 2008*, pages 1–9. IEEE, oct 2008. ISBN 9781424419357. doi: 10.1109/PHM.2008.4711414.

[59] E. Scanff, K. L. Feldman, S. Ghelam, P. Sandborn, M. Glade, and B. Foucher. Life cycle cost impact of using prognostic health management (PHM) for helicopter avionics. *Microelectronics Reliability*, 47(12): 1857–1864, dec 2007. ISSN 00262714. doi: 10.1016/j.microrel.2007.02.014.

[60] A. Sharma and H. Wehrheim. Testing monotonicity of machine learning models. 2020. ISSN 23318422.

[61] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou. Remaining useful life estimation - A review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1): 1–14, 2011.

[62] J. Sill. Advances in Neural Information Processing Systems (NIPS). 10:661–667, 1998.

[63] Richard S Sutton and Andrew G Barto. Reinforcement Learning: An Introduction Second edition, in progress. Technical report, 2018.

[64] Alan B. Tickle, Robert Andrews, Mostefa Golea, and Joachim Diederich. The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks. *IEEE Transactions on Neural Networks*, 9(6):1057–1068, 1998. ISSN 10459227. doi: 10.1109/72.728352.

[65] C. Vamo. Automatic algorithm for monotone trend removal. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 75(3):036705, mar 2007. ISSN 15393755. doi: 10.1103/PhysRevE.75.036705. URL https://journals-aps-org.tudelft.idm.oclc.org/pre/abstract/10.1103/PhysRevE.75.036705.

[66] C. Vamo and C. Maria. Automatic trend estimation. 2013. doi: 10.1007/978-94-007-4825-5. URL http://link.springer.com/10.1007/978-94-007-4825-5.

[67] M. Velikova and H. Daniels. Enforcing monotonicity of decision models: algorithm and performance. 2002.

[68] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3):586–600, may 2000. ISSN 10459227. doi: 10.1109/72.846731.

[69] Pingfeng Wang, Byeng D. Youn, and Chao Hu. A generic probabilistic framework for structural health prognostics and uncertainty management. *Mechanical Systems and Signal Processing*, 28:622–637, apr 2012. ISSN 08883270. doi: 10.1016/j.ymssp.2011.10.019.

[70] Xindong Wu. Top 10 algorithms in data mining. 14(1):1–37, jan 2008. ISSN 02191377. doi: 10.1007/s10115-007-0114-2.

[71] Oliver Wyman. Machine Learning: A Turning point for predictive maintenance? URL https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2017/sep/Predictive{_}Maintenance.pdf.

[72] Seungil You, David Ding, Kevin Canini, Jan Pfeifer, and Maya Gupta. Deep Lattice Networks and Partial Monotonic Functions. Technical report, 2017.

[73] Chun yu Zhao, Jing jiang Liu, Lun Ma, and Wei jun Zhang. Monotonicity evaluation method of monitoring feature series based on ranking mutual information. *Journal of Shanghai Jiaotong University (Science)*, 20(3):380–384, jun 2015. ISSN 19958188. doi: 10.1007/s12204-015-1641-8.

[74] Zeqi Zhao, Bin Liang, Xueqian Wang, and Weining Lu. Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Reliability Engineering and System Safety*, 164:74–83, aug 2017. ISSN 09518320. doi: 10.1016/j.ress.2017.02.007.

[75] Matej Žvokelj, Samo Zupan, and Ivan Prebil. Multivariate and multiscale monitoring of large-size low-speed bearings using Ensemble Empirical Mode Decomposition method combined with Principal Component Analysis. *Mechanical Systems and Signal Processing*, 24(4):1049–1067, may 2010. ISSN 08883270. doi: 10.1016/j.ymssp.2009.09.002.

A

Appendix

1. RUL Estimation process

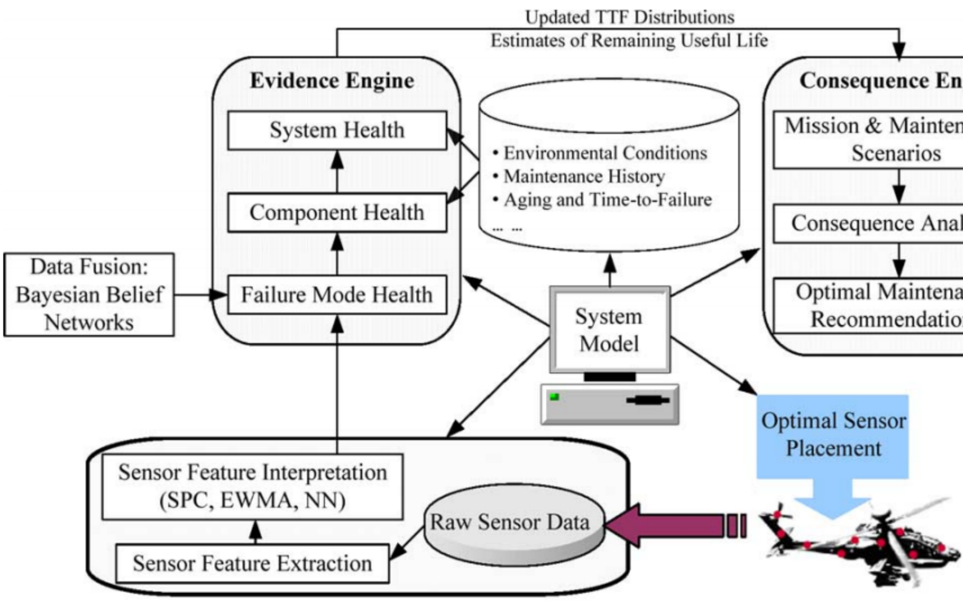

Figure A.1: SNL model which showcases the different engines that drive the RUL estimation process[51]

2. Dataset chosen for experimentation

The author chooses to develop a process that improves on the accuracy of prognostics of degradation of the component, it is essential to experiment with the data that have the final results (RULs in this case) for approval and understanding. Testing for such data is selected that replicates the real life scenarios and enables the algorithm to be used in a generalized format. Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) is a simulated data set used for replicating real-life commercial turbofan engines. The description of the engine model 90K is given below [21]:

- 90,000 lb thrust which is 400,340 N
- Atmospheric model altitude from sea level to 40,000 ft,
- Mach 0 to 0.9,
- Sea-level temperature -60 to 103 F

Engine control system [21]:

- Throttle-resolver angle (TRA) specific to a fan speed controller
- Three high-limit regulators prevented from exceeding its threshold limit for core speed, engine pressure ratio, High-pressure turbine (HPT) exit temperature
- Regulator to avoid static pressure at high pressure compressor (HPC) exit from going too low
- Moreover, to control the core speed, acceleration and deceleration limiters also exist

The data set inputs, output, and equilibrium values for all flight conditions are given in the Appendix A. The simplified turbo engine design is represented in Figure A.2 and the workflow of the engine with ducts and bleed omitted is seen in Figure A.3.

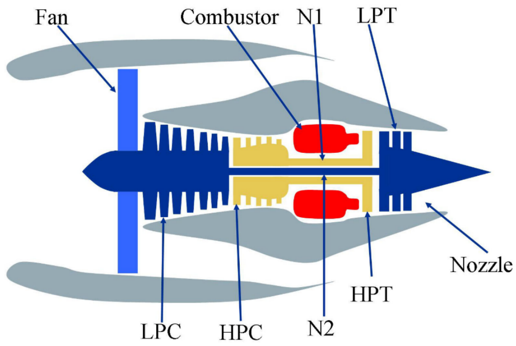

Figure A.2: A 90k turbofan engine, simplified version[21]

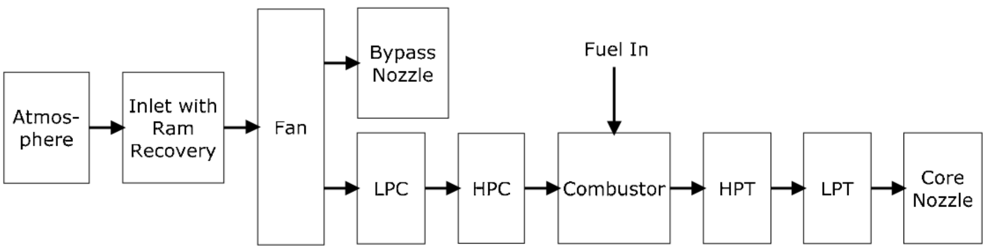

Figure A.3: Working of the engine divided in simulation boxes [21]

Data-set characteristics

The purpose of selecting the C-MAPSS data set was because it has varied characteristics and are very helpful to compute realistic prognostic models. These publications do not focus on the physics-of-failure of turbofan engines but describe the generation of these data sets and various practical aspects when using C-MAPSS data sets for prognostics. Characteristic of the data set is provided from [54][58] as follows:

1. To replicate a real system of the aircraft engine, a data of non-linear system of high fidelity is created for a multi-dimensional response
2. The data consist high level of noise
3. Fault conditions are in-cooperated in operational systems
4. Data of different level of complexities are given for training and testing the algorithms. The description of the data-sets is given in the Figure A.4

5. The data sets number 1-4 consists of training and testing data with increasing complexity. Already available separated data for analysis.

6. The ground truth RULs are also available to analyze the results of predictions

7. Number 5T and 5V are more complex and bigger data sets which were used for a competition. The RULs were not available at first, but after the completion of the competition, the results were made public

8. These data sets are designed for fault degradation analyses and hence fit perfectly in the authors experimental setup

Figure A.4: 90K turbofan degradation data-set types [54]

Datasets		#Fault Modes	#Conditions	#Train Units	#Test Units
Turbofan data from NASA repository	#1	1	1	100	100
	#2	1	6	260	259
	#3	2	1	100	100
	#4	2	6	249	248
PHM2008 Data Challenge	$\#5T$	1	6	218	218
	$\#5V$	1	6	218	435

Guidelines to process the data and build a fault degradation analysis is given in Figure A.5. This will be used as a reference for the process of developing a model.

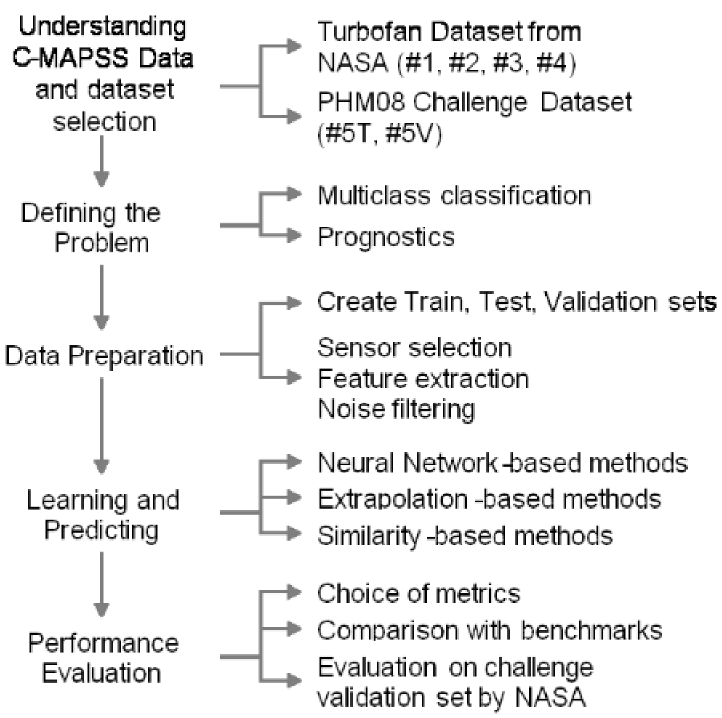

Figure A.5: Guidelines to use C-MAPSS data sets [54]

2.1. Dataset tables

Index	Name	Symbol
1	Fuel flow	Wf (pps)
2	Fan efficiency modifier	fan_eff_mod
3	Fan flow modifier	fan_flow_mod
4	Fan pressure-ratio modifier	fan_PR_mod
5	LPC efficiency modifier	LPC_eff_mod
6	LPC flow modifier	LPC_flow_mod
7	LPC pressure-ratio modifier	LPC_PR_mod
8	HPC efficiency modifier	HPC_eff_mod
9	HPC flow modifier	HPC_flow_mod
10	HPC pressure-ratio modifier	HPC_PR_mod
11	HPT efficiency modifier	HPT_eff_mod
12	HPT flow modifier	HPT_flow_mod
13	LPT efficiency modifier	LPT_eff_mod
14	HPT flow modifier	LPT_flow_mod

Figure A.6: Input to 90k [21]

Index	Symbol	Description	Units
1	Nf	Physical fan speed	rpm
2	Nc	Physical core speed	rpm
3	epr	Engine pressure ratio (P50/P2)	--
4	P21	Total pressure at fan outlet	psia
5	T21	Total temperature at fan outlet	°R
6	P24	Total pressure at LPC outlet	psia
7	T24	Total temperature at LPC outlet	°R
8	P30	Total pressure at HPC outlet	psia
9	T30	Total temperature at HPC outlet	°R
10	P40	Total pressure at burner outlet	psia
11	T40	Total temperature at burner outlet	°R
12	P45	Total pressure at HPT outlet	psia
13	T48	Total temperature at HPT outlet	°R
14	P50	Total pressure at LPT outlet	psia
15	T50	Total temperature at LPT outlet	°R
16	W21	Fan flow	pps
17	Fn	Net thrust	lbf
18	Fg	Gross thrust	lbf
19	SmFan	Fan stall margin	--
20	SmLPC	LPC stall margin	--
21	SmHPC	HPC stall margin	--
22	NRf	Corrected fan speed	rpm
23	NRc	Corrected core speed	rpm
24	P15	Total pressure in bypass-duct	psia
25	PCNfR	Percent corrected fan speed	pct
26	Ps30	Static pressure at HPC outlet	psia
27	phi	Ratio of fuel flow to Ps30	pps/psi

Figure A.7: List of 27 output variables and their units[21]

Symbol	Description	Units
accel_in	Accel limiter input	rpm/s
accel_out	Accel limiter output	rpm/s
BPR	Bypass ratio	---
DD	Decel limiter output	rpm/s
farB	Burner fuel-air ratio	---
far_HPT	HPT fuel-air ratio	---
far_LPT	LPT fuel-air ratio	---
Fdrag	Drag force	lbf
htBleed	Bleed enthalpy	
Nf_dot	Fan acceleration	rpm/s
Nc_dot	Core acceleration	rpm/s
Nf_dmd	Demanded fan speed	rpm
P2	Pressure at fan inlet	psia
PCNfRdmd	Demanded corrected fan speed	pct
PCNfR_filtered	Output of pcnfr filter for gain scheduling	pct
PR_HPC	Pressure ratio of HPC	---
PR_HPT	Pressure ratio of HPT	---
PR_LPT	Pressure ratio of LPT	---
tau_HPC	Torque of HPC	ft-lb
tau_HPT	Torque of HPT	ft-lb
tau_LPT	Torque of LPT	ft-lb
TRA	Throttle resolver angle	deg
T2	Total temperature at fan inlet	°R
W22	Flow out of LPC	lbm/s
W25	Flow into HPC	lbm/s
W31	HPT coolant bleed	lbm/s
W32	HPT coolant bleed	lbm/s
W48	Flow out of HPT	lbm/s
W50	Flow out of LPT	lbm/s
Wf_dot	Derivative of fuel flow	lbm/s^2
x1,…,x5	Solver outputs	

Figure A.8: Non-output variables[21]

Name	Alt, ft	Mach	Tsl, °F	TRA, deg	Fuel flow, pps	Fan speed, rpm	Core speed, rpm	epr	HPT outlet temp, °R	Net Thrust, lbf
FC01	0	0	59	100	6.835	2388	9051	1.300	2072	86,336
FC02	0	0.25	59	100	7.085	2403	9084	1.261	2083	66,755
FC03	0	0.25	86	96	7.043	2432	9274	1.247	2162	64,250
FC04	1000	0	59	100	6.567	2380	9021	1.300	2059	83,293
FC05	10 K	0.25	59	100	4.661	2319	8774	1.259	1947	45,830
FC06	20 K	0.70	59	100	3.863	2324	8719	1.077	1909	25,774
FC07	25 K	0.62	59	60	1.670	1915	8006	0.938	1534	11,475
FC08	35 K	0.84	59	100	2.120	2223	8346	1.024	1750	13,552
FC09	42 K	0.84	59	100	1.518	2212	8317	1.023	1744	9,647
FC10	0	0	59	80	5.511	2224	8837	1.227	1941	71,652
FC11	0	0	59	60	4.254	2028	8592	1.165	1792	57,181
FC12	0	0	59	40	3.075	1797	8299	1.114	1623	42,562
FC13	0	0	59	20	2.013	1497	7946	1.073	1433	28,016
FC14	0	0	59	0	1.123	1146	7503	1.044	1214	13,448

Figure A.9: Equilibrium values[21]

B

Appendix

1. CNN-LSTM model on test units FD001 dataset

This model is used for validating the results of the experiment carried out with the training dataset. For this model, the entire training dataset is trained and the test FD001 is used for testing the model. RUL FD001 has the final values which is used for validating the process. In this experiment, the entire values of test datasets are not available, only the first few sensor values for each engine is available. Therefore, all the visual analysis carried out in the paper cannot be carried out in this experiment. The 4 preprocessed datasets are again used for comparison. However, the test dataset is not converted by any preprocessing step. This step would be better for evaluating the right accuracy as the test dataset would resemble the real-life scenario. Now, after implementing the 4 different data types at three different epochs, the following results in the Table B.1 were seen.

Table B.1: Numerical evaluation of each experimentation step.

Datasets	5 Epochs				50 Epochs			
	MAE	RMSE	R2	Score	MAE	RMSE	R2	Score
Raw	11.57	17.15	0.75	7.34e4	18.47	26.89	0.81	3.23e5
EMV	11.27	16.42	0.72	1.17e5	19.20	25.45	0.78	3.42e5
Rolling mean	13.24	18.45	0.79	1.81e5	12.47	21.29	0.84	2.01e5
ACD	**10.66**	**14.47**	**0.81**	**7.08e4**	**10.75**	**17.40**	**0.87**	**1.41e5**

At a first glance of the table, one can already say that the ACD dataset has outperformed its competition. The error score for each epoch has shown improvement. It is also worth noticing that the result of 5 epoch category has performed better than that of the 50 epochs. Therefore, more iterations of the data do not necessarily improve the predictions.

With a complex algorithm which improves on underfitting, it essentially improves on the pitfalls of monotonic constraints. It can also be seen visually in Figure B.1, where the histogram chart of the ACD dataset shows the least amount of errors on the late prediction side. There is a bigger peak in the center and followed by a smaller peak in the early prediction side. Early prediction, which is not very far from the actual RUL, is still acceptable. It would not result in a big loss. Whereas for the raw and EMV, the results show that most of the predictions were made early and quite a few were also made later than the actual values. The combination of the ACD dataset and the CNN-LSTM model worked well and the results can be seen. One of the reasons for it to work so well is because the neural network model develops a piecewise linear curve for predicting RUL. As the ACD filter would have already developed a monotonic dataset, the ML model manages to learn faster and produce predictions accurately.

Furthermore, comparing the two models and their numerical results, the neural network performed significantly better. Moreover, the RF regressor was trained and tested on the same dataset. While it was not the same for the complex model. A direct comparison can not be made, although with the predicted results, it proves that

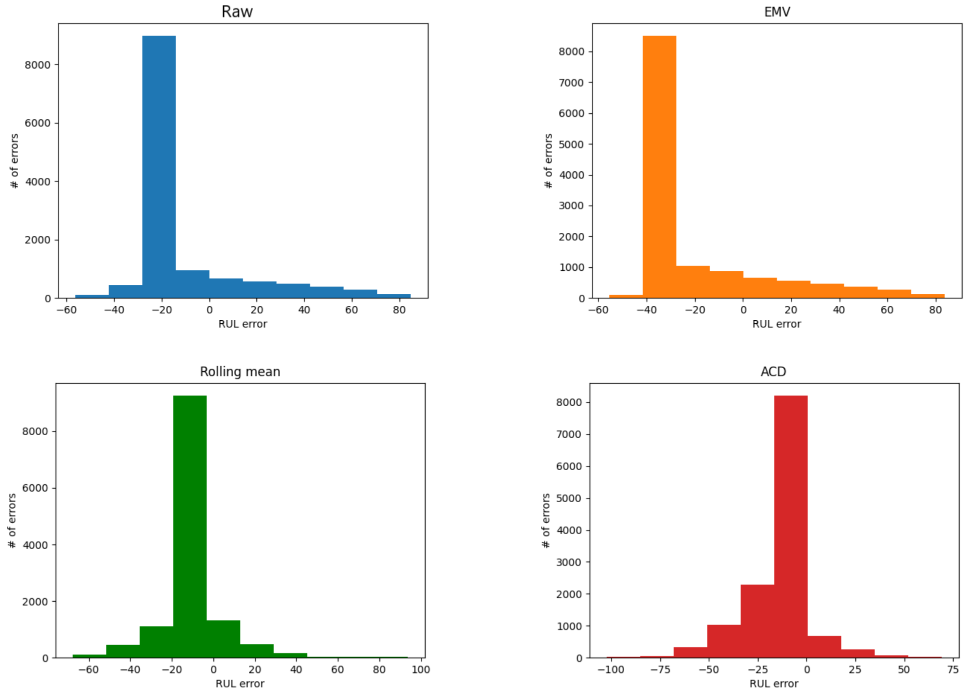

Figure B.1: Error distribution function to identify if failures are predicted early or late

using a monotonic constraint for fault progression data would improve the prediction and accuracy of the model. The impact of producing more accurate predictions is on the cost of predictive maintenance. Accurate maintenance could be called for and self-healing materials can also have a longer life without hinderence to maintenance operations.