# Data-driven Approach for Fault Prognostics of Industrial Systems - From Using No, Insufficient, to Multiple Historical Degradation Sequences

Koceila Abid

IMT Lille Douai
École Mines-Télécom
IMT-Université de Lille

Université de Lille

# THESE

Présentée dans le but d'obtenir le grade de

# DOCTEUR

Spécialité : Automatique, Génie Informatique, Traitement du Signal

par


## Koceila Abid


Doctorat de l'université de Lille délivré par IMT Lille Douai

Titre de la Thèse:

## Data-driven Approach for Fault Prognostics of Industrial Systems - From Using No, Insufficient, to Multiple Historical Degradation Sequences

Soutenue le 25 Novembre, 2020 devant le jury d'examen :

**Président,** Sylvain Lecomte, Professeur à l'IMT Lille Douai

**Examinatrice,** Latifa Oukhellou, Directrice de Recherche à l'IFSTTAR

**Rapporteur,** Kamal Medjaher, Professeur à l'ENI de Tarbes

**Rapporteur,** Dimitri Lefebvre, Professeur à l'Université Le Havre

**Directeur de Thèse,** Moamar Sayed-Mouchaweh, Professeur à l'IMT Lille Douai

**Encadrante de Thèse,** Laurence Cornez, Ingénieure Chercheuse au CEA

**Membre invité,** Pascal Fugier, Ingénieur Chercheur au CEA

Laboratoire d'accueil:

CERI Science du Numérique, UR Informatique et Automatique
Ecole doctorale Science pour l'ingénieur (ED SPI)

This manuscript is the result of the research work supported by the French Alternative Energies and Atomic Energy Commission (CEA), The Region of Hauts de France, European Regional Development Fund, European Union.

# Acknowledgments

Throughout the writing of this dissertation I have received a great deal of support and assistance.

I would first like to thank the CEA and the "Région Hauts de France" for the funding of my thesis work. I am also thankful to IMT Lille Douai in particular "CERI Science du Numérique" for the warm welcome in their offices.

I would like to thank Professor Sylvain Lecomte to be president of the jury. I also want to thank Professor Kamal Medjaher and Professor Dimitri Lefebvre for reviewving my thesis manuscript. I gratefully thank Mrs Latifa Oukhellou for examining my work. I greatly appreciated the discussions and the relevant remarks during my defense.

I wish to express here my deep gratitude to my supervisors, Professor Moamar Sayed-Mouchaweh and Mrs Laurence Cornez, for their patient support. Their insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. In addition, I would like to thank my CEA referent Pascal Fugier for the precious help and the follow-up of the thesis project.

I would like to acknowledge all my colleagues and friends at IMT Lille Douai, without forgetting all my colleagues at CEA Hauts de France. I would like to thank all my friends who supported me during three years, in particular Sido, Mohamed, Hicham, Amine, Hajer, Sameh, and Roza.

Finally, a huge thanks to my sisters and parents because I could not have completed this dissertation without their moral support especially my mother. My parents were always my inspiration and example in life.

***" This success is dedicated to my parents "***

**K. Abid**

# Contents

# List of Figures

# List of Tables

# List of abbreviations

**AI** Artificial Intelligence

**ANN** Artificial Neural Network

**AR** Auto Regressive

**ARMA** Auto Regressive Moving Average

**BLSTM** Bidirectional Long Short Term Memory

**BPFI** Ball Pass Frequency Inner

**BPFO** Ball Pass Frequency Outer

**C-MAPSS** Commercial Modular Aero-Propulsion System Simulation

**CBM** Condition Based Maintenance

**CNN** Convolutional Neural Network

**CRA** Cumulative Relative Accuracy

**EOL** End Of Life

**FFT** Fast Fourier Transform

**GLM** Generalized Linear Model

**GPR** Gaussian Process Regression

**HHT** Hilbert Huang Transform

**HI** Health Indicator

**HMM** Hidden Markov Model

**IMF** Intrinsic Mode Function

**ISOMAP** Isometric Mapping

**KL** Kullback-Leibler divergence

**KNN** K Nearest Neighbor

**LSTM** Long Short Term Memory

**MA** Moving Average

**MAE** Mean Absolute Error

**MAPE** Mean Absolute Percentage Error

**MLP** Multi Layer Perceptron

**OCSVM** One Class Support Vector Machine

**PCA** Principal Component Analysis

**PHM** Prognostics and Health Management

**PSD** Power Spectral Density

**RA** Relative Accuracy

**RBF** Radial Basis Function

**RMS** Root Mean Square

**RMSE** Root Mean Square Error

**RNN** Recurrent Neural Network

**RUL** Remaining Useful Life

**SK** Spectral Kurtosis

**STD** Standard Deviation

**STFT** Short Time Fourier Transform

**SVM** Support Vector Machine

**SVR** Support Vector Regression

**WPD** Wavelet Packet Decomposition

# Chapter 1

# General introduction

## 1.1 Context and motivations

Major economic losses in industrial companies in different domains (e.g., energy, manufacturing and automobile, transportation, etc.) are due to repairing and downtime costs that result from unforeseen equipment failure. Repairing costs for nuclear plants can exceed 600 million$ without including lost revenues due to the downtime (J. B. Coble 2010a). System downtime can also lead to huge economic losses, an unexpected one day stoppage of production in industry may cost up to 100,000 € for coal power station and chemical factory, 200,000 € for Pulp and Paper, and can reach up to 300,000 € for nuclear plant (Helle 2006), as shown in Figure 1-1.

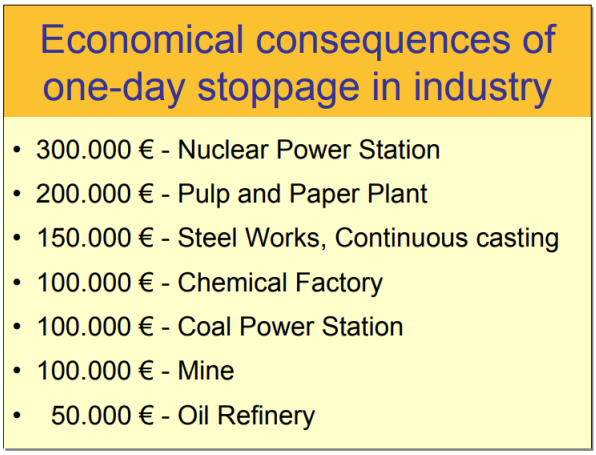



Figure 1-1: Economical losses for one day stoppage in industry (Helle 2006)

Traditional maintenance strategies include two categories: corrective and preventive maintenance. Corrective maintenance aims to repair the system only when a failure has occurred. The process of the corrective maintenance strategy has the advantage to avoid any unnecessary maintenance actions by only repairing the failed component. However, it is very costly because of the fault consequences on the system safety, availability and reliability. On the other hand, preventive maintenance aims to schedule periodic inspections of the system in order to prevent failures and their consequences. Its cost is very high because the replacement of critical components is scheduled in period interval regardless of the current health conditions. An alternative to those traditional maintenance strategies is the predictive maintenance.

Prognostics and health management (PHM) is an advanced maintenance strategy that can overcome the limits of traditional maintenance strategies. PHM permits to monitor the health conditions of the system and trigger the maintenance actions only when needed. It is a cost-effective strategy comparing to the traditional strategies since it can reduce the repairing costs of the corrective maintenance as well as the costs of periodic inspections generated by the preventive maintenance. Generally, the pipeline of the PHM strategy includes five steps: data acquisition, data processing, fault diagnostics, fault prognostics, and decision support (health management). Fault prognostics is one of the main steps for achieving PHM strategy, which aims to estimate the Remaining Useful Life (RUL) before failure. It can help to plan the maintenance actions in advance before failure occurrence in order to avoid systems downtime and reduce the revenue losses.

## 1.2 Objectives and contributions

Fault prognostics based on data-driven approaches is efficient, especially when no physical or mathematical model about the operation of the system is available, or when it is challenging to build a physical model where several components are in interaction. Fault prognostics based on data-driven approaches require several historical degradation sequences in order to construct a model that can achieve a desirable RUL

estimation accuracy. Different approaches are proposed in the literature for RUL estimation when several historical degradation sequences are available (Medjaher et al. 2012; Soualhi et al. 2014; X. Li et al. 2018; Heimes 2008). However, in industrial systems, these historical sequences are often unavailable (e.g., new machines) or insufficient (i.e., few available sequences that do not cover all the degradation evolution dynamics or conditions). In order to overcome this main challenge, three data-driven approaches are proposed for RUL estimation when no degradation sequences, few sequences, and multiple sequences are available a priori, as it is shown in Figure 1-2.

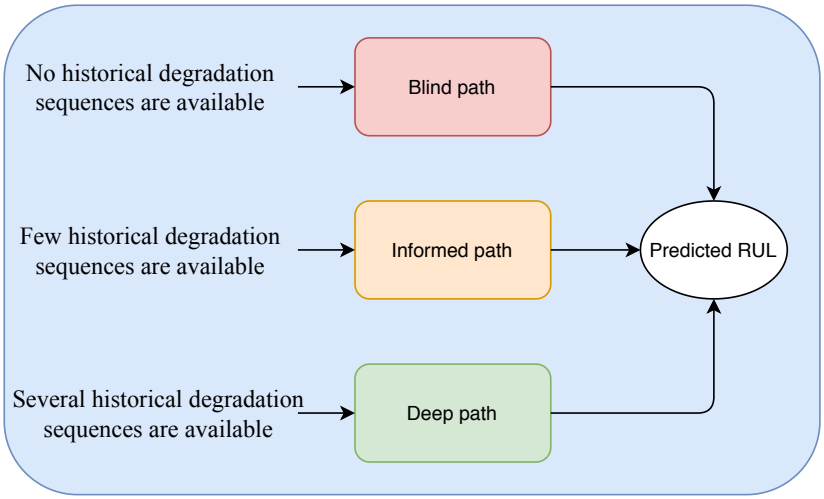



Figure 1-2: Three proposed approach for fault prognostics according to the available degradation sequences

The general proposed approach includes three RUL estimation paths: blind, informed, and deep paths depending on the number of available degradation sequences. The blind path is triggered when no a priori sequences are available, it aims to estimate the RUL using an adaptive model where the prediction becomes more accurate over time with the arrival of new incoming degradation data. The informed path is triggered when one or more a priori sequences are available, the RUL is estimated using several adaptive models, and the more the number of a priori sequences is increasing the more the RUL estimation is accurate and reliable. The deep path is used when multiple a priori sequences are available, where the RUL estimation is more efficient and accurate compared with the other paths. More details about these paths are provided in the next section.

## 1.3 Manuscript organization

The thesis manuscript is organized as follows:

**Chapter 2 - State of the art of fault prognostics for the predictive maintenance.** This chapter presents a review about the state of the art techniques used for the prognostics and health management strategy pipeline, especially the prognostics step. First, the evolution of the maintenance strategies is shown from corrective, preventive, until predictive maintenance. Then, the pipeline of the PHM strategy is presented, including Health Indicator (HI) construction, degradation detection, and RUL estimation. The different techniques for constructing the HI are described as well as the metrics able to select the best one automatically. Thereafter, the degradation detection methods are presented, and classified according to how they detect a degradation: it can be detected using normal and faulty data about the system operation or using only normal data. Finally, the RUL estimation approaches are described, and their performances are compared in order to justify the use of data-driven approaches. The chapter focuses on the latter category of approaches in order to estimate the RUL.. The chapter ends with a discussion about the previously described techniques that permit to achieve fault prognostics efficiently.

**Chapter 3 - Prognostic approach with insufficient a priori degradation sequences.** This chapter presents the data-driven based approach for achieving fault prognostics when there is insufficient (no or too few) historical degradation sequences. First, the HI library is defined, then the degradation is detected using only normal data of the system operation by applying a One Class Support Vector Machine (OCSVM). When the degradation is detected, the blind path or the informed path is triggered depending on the availability of a priori sequences. The blind path is triggered when no a priori sequences are available, where the HI is selected dynamically at each time cycle, and an adaptive model is used for the HI extrapolation until failure. The informed path is triggered when some a priori sequences are available (at least one sequence is available). It aims at predicting RULs using different adaptive HI-model pairs, where the final RUL is deduced by fusing the computed RULs. This

proposed approach is validated using vibration data collected from a real degraded bearing of a wind turbine high speed shaft, vibration data generated from a degraded bearing with inner and outer race fault, and sensors data collected from a degraded aircraft engine (C-MAPSS dataset).

**Chapter 4 - Prognostic approach with multiple a priori degradation sequences.** This chapter presents a data-driven based approach for achieving fault prognostics when several historical degradation sequences are available. The proposed approach is based on a deep ensemble method by combining the RUL estimation of two deep learning models that have proven their effectiveness for fault prognostics, named Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). CNN architecture can automatically extract relevant information (features) by applying several convolution filters on the raw data, while the LSTM has the ability to capture the sequential information in time series data. Two RULs are predicted with each model, where the final RUL is obtained by fusing the predicted RULs. The proposed deep ensemble approach for RUL estimation is validated using a filter clogging dataset available for the PHM Europe data challenge, and C-MAPSS dataset. It achieved promising performance compared with the state-of-the-art results.

**Chapter 5 - General conclusion.** This chapter summarizes the proposed contributions, then presents the open issues and future directions in order to improve the proposed fault prognostics approaches.

## 1.4 Publications

- Abid, K., Sayed-Mouchaweh, M., & Cornez, L. (2020). Adaptive Fault Prognostic Approach for the Remaining Useful Life Estimation of Bearings by Dealing with the Lack of Historical Degradation Data. In IET Science, Measurement & Technology. (Submitted)

- Abid, K., Sayed-Mouchaweh, M., & Cornez, L. (2020). Adaptive Data-driven Approach for the Remaining Useful Life Estimation when Few Historical Degradation Sequences are Available. In International Conference on Machine Learning and Applications, IEEE. (Accepted)

- Abid, K., Sayed-Mouchaweh, M., & Cornez, L. (2020). Deep Ensemble Approach for RUL Estimation of Aircraft Engines. In Mediterranean Forum of Data Science MEFDATA2020, Springer. (Accepted)

- Abid, K., Sayed-Mouchaweh, M., & Cornez, L. (2019, September). Adaptive Machine Learning Approach for Fault Prognostics based on Normal Conditions-Application to Shaft Bearings of Wind Turbine. In Annual Conference of the PHM Society (Vol. 11, No. 1).

- Abid, K., Sayed-Mouchaweh, M., & Cornez, L. (2018, September). Fault prognostics for the predictive maintenance of wind turbines: State of the art. In Joint european conference on machine learning and knowledge discovery in databases (pp. 113-125). Springer, Cham.

# Chapter 2

# State of the art of fault prognostics for the predictive maintenance

## 2.1 Introduction

Operation reliability and availability of industrial systems influence on the profit generation and competitiveness of industrial companies. With the upgrade of the production equipment and automation level, industrial companies have experienced a steady increase in maintenance costs. This implies the importance of applying a cost-effective maintenance strategy that can maintain the reliability and availability while reducing downtime and production loss costs of machinery, process, and other production means of industrial systems.

In this chapter, the different maintenance strategies are described. Then, the pipeline of the prognostics and health management strategy is presented, and its different steps required to perform fault prognostics are investigated. The main fault prognostic steps include health indicator construction, degradation detection, and remaining useful life estimation. The chapter ends with a discussion highlighting the advantages of the suitable techniques that can be used for our work in the next.

## 2.2 Maintenance strategy evolution

In order to plan maintenance actions, three basic maintenance strategies can be defined: corrective maintenance, preventive maintenance, and condition based maintenance. The application of these strategies evolved with time in order to reduce the lifecycle costs of systems (Kim et al. 2016).

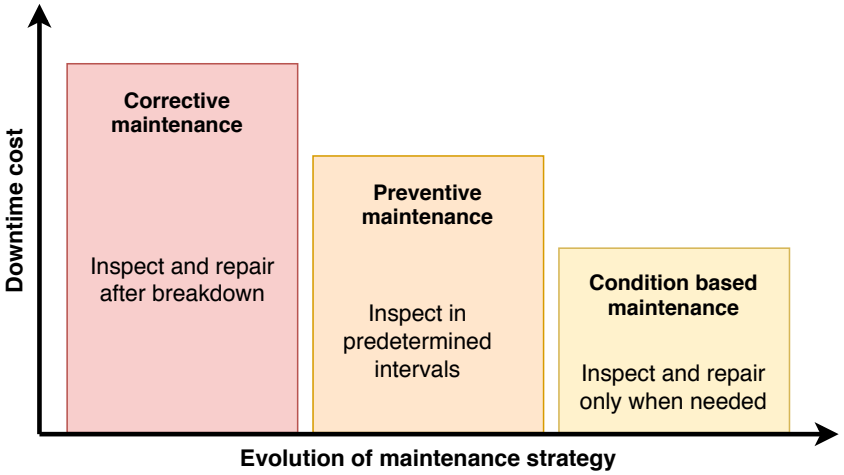



Figure 2-1: Evolution of maintenance strategy

Figure 2-1 presents the evolution of maintenance strategy in order to reduce the downtime costs. The maintenance strategy evolved from corrective maintenance and preventive maintenance to the condition based maintenance or predictive maintenance. The principle of the corrective maintenance strategy is to run the system until it fails (breaks), the preventive maintenance schedules inspections of the system in predetermined intervals, while the condition based maintenance helps to monitor the system in real time and recommends maintenance actions only when needed. The three strategies are detailed next.

### 2.2.1 Corrective maintenance

Corrective maintenance, also called breakdown, reactive, or unplanned maintenance, is the earliest used maintenance strategy. There are no routine maintenance tasks to perform, and it aims to repair the system only when the useful life of the system is

consumed (after failure occurrence). Corrective maintenance is suitable when equipment shutdowns do not affect product quality or revenue generation, or when repair and downtime time costs are within an acceptable range. However, this strategy takes the longest time to start repairing actions, especially when replacement parts are not available because there is no time to prepare the maintenance actions in advance (before failure). This strategy leads to a high costs because downtime events are often unplanned, more frequent, and longer in duration.

### 2.2.2 Preventive maintenance

Preventive maintenance, also called time-based maintenance or planned maintenance, aims to schedule periodic inspections of the system in order to prevent failures and their consequences. In this strategy, key elements have a prescheduled replacement interval, regardless the current health conditions. It is a cost-effective strategy if all parts are expected to fail in the same time, which is often not the case in real systems. The maintenance costs increase in this strategy because it replaces all parts even if many of them may not need to be replaced. Moreover, the scheduled inspection requires intrusion on equipment which increases the downtime costs. While this approach can help to reduce system failure and increase the residual life, the process is labor-intensive, the inspection requires to downtime the system, and the inspection is time-dependent regardless the condition of the system which increases the maintenance cost.

### 2.2.3 Condition based maintenance

With the development of technology, modern industrial systems became more complex entailing to increase their potential to fail. Therefore, their maintenance costs, especially preventive maintenance, become very expensive. In order to reduce maintenance costs while maintaining system reliability and safety, condition based maintenance became a promising solution for industrial systems.

The condition based maintenance (CBM), also called condition directed mainte-

nance or predictive maintenance, which is applied to overcome the limits of preventive maintenance. While the preventive maintenance is time dependant regardless of the system health conditions, the CBM is dependent on the current health conditions of the system. CBM permits to take actions by repairing or replacing the degraded parts before product quality is reduced to unaccepted levels or failures occurred. Even if CBM requires investment costs in order to implement, operate, and maintain, those costs remain lower than ones generated by production losses due to downtime (Niu 2017). The predictive maintenance is efficient when the degradation dynamics of the system is known (not random) and evolves with time, and when measurable parameters about the system conditions are available (collected by sensors). Moreover, the CBM is not intrusive on the system's components and does not require system downtime for inspection since the inspection actions are triggered only when degradation is detected.

## 2.3 Prognostics and health management strategy

In the literature, more and more works are investigating the Prognostics and Health Management (PHM) strategy, which is a strategy that can help to achieve the condition based maintenance (predictive maintenance). PHM can enhance the predictive maintenance by evaluating the current health state of the system, detect and diagnose the incipient fault, and determine how long from now a failure will happen in a system given the current operating conditions (Das et al. 2012).

PHM is a strategy that focuses more on incipient fault detection, current health assessment and remaining useful life prediction (Lee et al. 2014). PHM strategy is applied for several goals: predicting failure in advance, minimizing the number of unscheduled maintenance, increasing the availability of the system, reducing the maintenance costs by decreasing the inspection costs, decreasing downtime costs, and optimizing the maintenance actions. According to Jardine et al. (2006), PHM program consists of three main steps including data acquisition step in order to collect relevant data for the system health monitoring, data processing step where the col-

lected data are analyzed by using advanced methods for achieving fault diagnosis and prognosis, and the maintenance decision making step in order to recommend optimal maintenance actions. Callan et al. (2006) proposed a PHM architecture with five steps: Data Manipulation, Condition Monitoring, Health Assessment, Prognostics, and automatic decision reasoning. Four main steps for PHM strategy is presented by (Kim et al. 2016) including data acquisition step (collect condition monitoring data and extract features), diagnostics step (what is the fault and how severe is it?), prognostics step (what is the remaining useful life?), and finally health management step (optimal management on maintenance and logistics). To sum up, Figure 2-2 shows the main steps to perform PHM strategy.

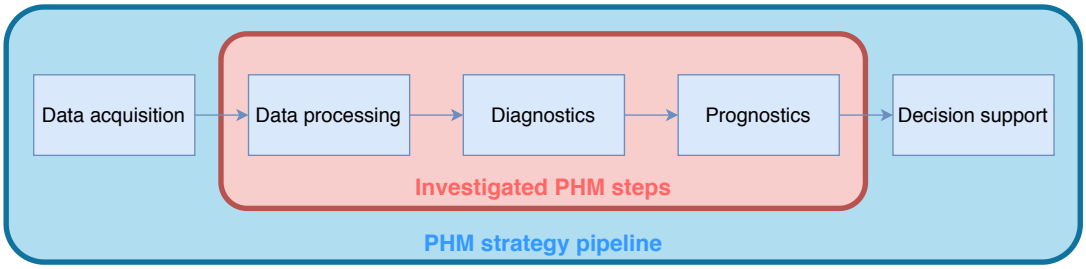



Figure 2-2: Main steps of the PHM strategy

Data acquisition step is used to collect relevant data from sensors that are placed on critical components, such as bearing in rotating machines that undergo slow degradation, which enable performing the prognostics (estimating their RUL) that may reduce their maintenance costs significantly. In the data processing step, the collected data are analyzed in order to extract significant indicators (features) about the system's health evolution useful for the next steps. The diagnostics step permits to achieve incipient fault detection, fault isolation or localization. Degradation detection triggers the prognostics step where the RUL is estimated. The final step is the decision support step, and it uses the obtained information from the previous steps (health state, root cause, and RUL) in addition to other information (e.g., priority, logistics) in order to recommend the optimal maintenance actions. This thesis focuses on data processing, diagnostics and prognostics steps of the PHM strategy, in particular prognostics, as it is illustrated in Figure 2-2 in the red rectangle.

## 2.4 Health indicator construction

In order to detect an incipient fault (e.g., wear or crack) in its early stage and follow its progress over time to be a failure, indicators characterizing the system's, or one of its components, health state, as well as its dynamic evolution over time, must be built. To this end, it is necessary to apply condition monitoring by using data collected from sensors. The collected data contains raw information about the system health state. Hence, in general the collected data requires careful processing to build a suitable, sensitive, health indicators.

The Health Indicator (HI) construction is one of the most important steps to achieve prognostics. It can represent the system performance (e.g., produced energy for a wind turbine), or it can represent the system operation conditions (e.g., temperature, vibration, oil debris density). In general, the HI permits to follow the health state of a component or a system over time. When the HI evolution starts increasing (or decreasing) over time, a drift appears from nominal (normal) operating conditions towards failure as illustrated in Figure 2-3.

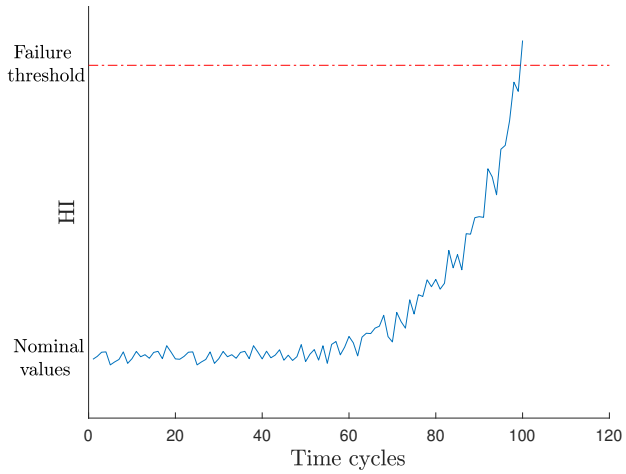



Figure 2-3: Illustration of a HI evolution in the degradation occurrence

The HI should be sensitive to the degradation dynamics and can be built based on the extraction and processing of one feature, or can be based on fusing multiple features extracted from the monitoring signals. The HI based on a single feature can be constructed using the collected raw measurements, generated residuals between

nominal and incoming measurements, or signal processing techniques such as time, frequency, or time-frequency domain signal processing. The second category is based on a fusion of multiple features by merging different HIs in the aim to exploit their diversity (Abid et al. 2018; Y. Lei et al. 2018). This merge can be achieved based on dimension reduction (e.g., principal component analysis), based on a distance from the normal or faulty class (e.g., Euclidean and Mahalanobis distance), or it can be based on the fusion using regression techniques (e.g., support vector regression or artificial neural network). However, this kind of HIs loses physics meaning as well as their interpretability. The two categories are detailed in the next. Figure 2-4 shows the classification of the different HI construction methods.

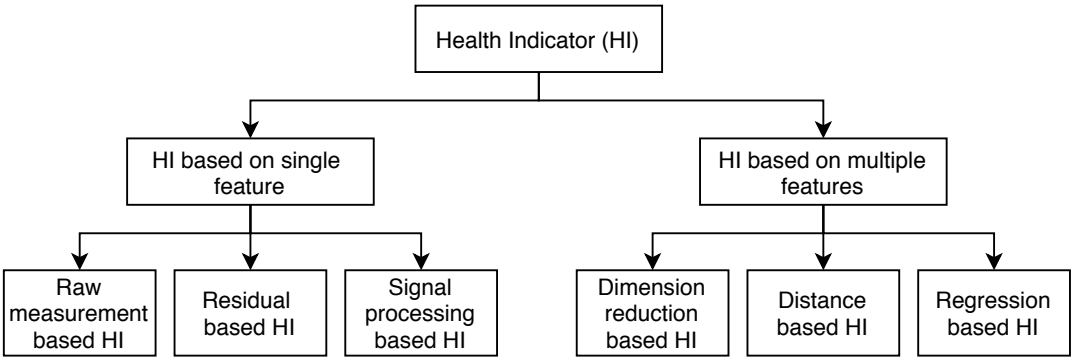



Figure 2-4: Classification of HI construction techniques

### 2.4.1 Health indicator based on a single feature

In this category, the health state of the system is monitored using only one HI (i.e., one feature). This feature could be chosen manually by a human expert or automatically by using HI selection criteria proposed in the literature. The HI based on a single feature can be based on raw measurements, residuals, or signal processing techniques.

#### Raw measurement based HI

The HI can be based on collected raw measurement from the monitored system or component. For instance, in (Dupuis 2010), the HI is constructed using the full signal of oil debris monitoring located on wind turbine gearbox, which is sensitive to bearing

spall. J. Zhu et al. (2013) used the viscosity and dielectric raw signals as HI in order to monitor the lubrication oil degradation of the wind turbine gearbox.

### Residual based HI

The HI based on a residual is computed as the deviation between the current new measurement and the nominal operation conditions of the system. In the work of Uluyol et al. (2011), a residual is computed between the current generated power and the power curve given by the manufacturer of wind turbines. The residual is calculated in order to predict the health state of the system based on its performance. A residual between a nominal reference temperature and the measured temperature of wind turbine bearing is computed in (Bangalore et al. 2015), while the nominal (reference) temperature is predicted using artificial neural network (ANN).

### Signal processing based HI

It is necessary to use signal processing techniques to deal with high sampling frequency signal (e.g., vibration signals), because it is difficult to clearly observe the degradation start as well as its evolution, due to the variation in operating conditions and the effect of noise.

In the literature, most of the prognostic works construct the HI using signal processing techniques (time, frequency, and time-frequency domain features). In general, they apply traditional time domain techniques such as Root Mean Square (RMS), kurtosis, skewness, peak value, shape factor, crest factor, impulse factor...etc. RMS is widely used as HI for health monitoring (Y. Li et al. 1999; Y. Lei et al. 2016a; Ahmad et al. 2017). In (Y. Lei et al. 2016a), RMS is computed using vibrations signal in order to monitor the health state of bearings and estimate the remaining useful life. The RMS value is also computed in (Ahmad et al. 2017) in order to estimate the RUL of bearings, where the RMS has shown a trend that can be used to characterize different health states of a bearing (normal, incipient, and severe stages). In the work of Saidi et al. (2017), the kurtosis is used to monitor the degradation of a high speed shaft bearing of a wind turbine. This HI is computed from 50 days run-to-failure vi-

bration measurements (see Figure 2-5). This figure shows an increasing trend of the HI over time due to the degradation severity evolution. Javed et al. (2014) proposed new features based on trigonometric functions computed from the vibration signals, then cumulative transformation is performed to enhance the trend of the degradation evolution.

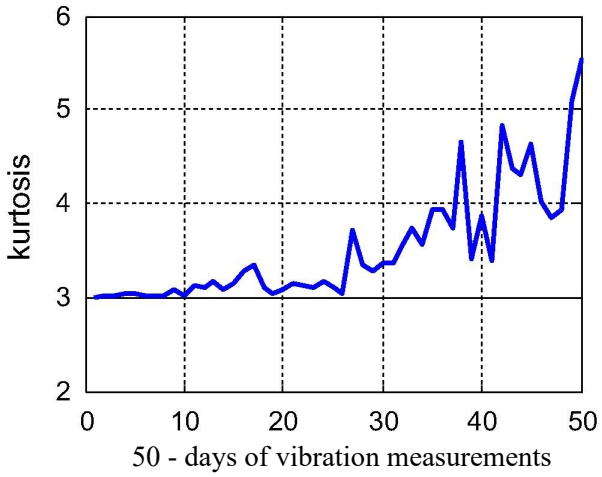



Figure 2-5: HI based on a single feature (Saidi et al. 2017)

Frequency domain techniques are efficient for HI construction from vibration signals in order to detect degradations and predict the RUL of mechanical components (e.g. bearings). Fast Fourier Transform (FFT) is widely used for feature extraction on vibration signals. FFT decomposes the vibration signal into its spectrum, which contains frequency components and their amplitudes. FFT is used in (Liao 2013) for features extraction in order to predict the RUL of a spindle test-bed machine and rolling bearing. Z. Zhang et al. (2013) transformed the pre-processed vibration signals into frequency domain signals using FFT for HIs construction, where the latter are used as input for ANN in order to achieve fault diagnosis and prognosis.

Spectral Kurtosis (SK) has also proven its effectiveness for HI construction in order to perform fault detection and prognosis of bearings using vibration signals (J. Tian et al. 2015; Saidi et al. 2017). SK is the kurtosis of the spectral components, it can detect impulsive bearing signatures. J. Tian et al. (2015) applied SK for fault detection of motor bearing, while it is applied in (Saidi et al. 2017) for fault detection and prognosis of high speed shaft bearing of a wind turbine. In (Cheng et al. 2017),

current signal is analysed to extract HI by using the Power Spectral Density (PSD) of the current signal.

Some works use time-frequency techniques for HI construction, including Short Time Fourier Transform (STFT), Wavelet Packet Decomposition (WPD), and Hilbert Huang Transform (HHT). WPD is used for signal analysis, it aims to decompose the frequency domain of the signal into several frequency subsets. In (Malhi et al. 2011), RMS and peak values are computed on the wavelet coefficient and were chosen as HI for the RUL estimation. Tobon-Mejia et al. (2012) and Z. Zhang et al. (2013) used WPD for HI construction, where it is applied on vibration data of bearings for RUL estimation. HHT can decompose a signal into so-called Intrinsic Mode Functions (IMF), where the IMF are used to compute the instantaneous frequency data. It is applied in (Soualhi et al. 2014; Hui Li et al. 2010) to extract time-frequency HIs, where the computed HIs are used for fault detection, diagnosis and prognosis.

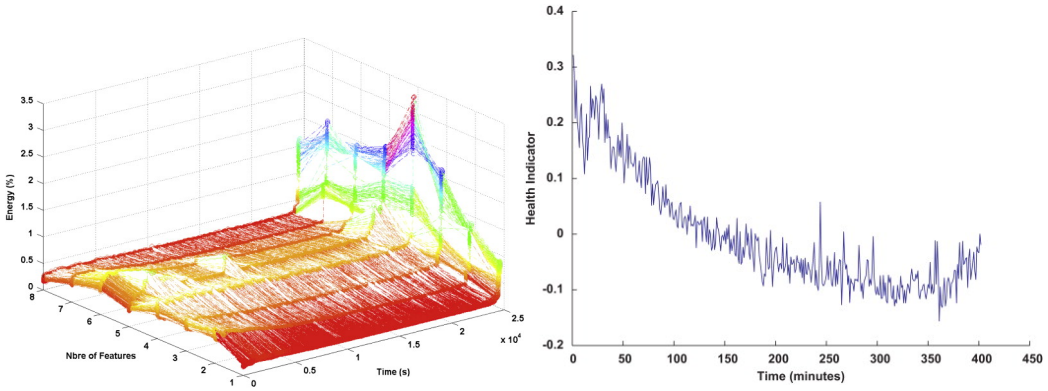
### 2.4.2 Health indicator based on multiple features

In some cases where the degradation dynamic evolution is complex (different operating conditions, noises etc.), it can be useful to build a HI issued from the combination of several features in order to exploit their diversity and complementarity. This combination can be obtained using different techniques such as dimension reduction, Distance-based, or regression techniques. However, this fusion entails the loss of physics meaning (loss of interpretability). This fusion is mainly used for fault detection and diagnosis (Toubakh et al. 2016; Soualhi et al. 2014), as well as in some works for fault prognostics (Benkedjouh et al. 2013; Chammas et al. 2015).

#### Dimension reduction based HI

These techniques can merge several features into one HI by using dimension reduction techniques such as Principal Component Analysis (PCA) or Isometric Mapping (ISOMAP). Benkedjouh et al. (2013) extracted eight features from a vibration signal using WPD. Each feature represents the level of energy at each level of decomposition.

These features are then merged in order to construct one HI, as illustrated in Figure 2-6. This fusion is achieved by using a nonlinear dimension reduction techniques called ISOMAP (Tenenbaum et al. 2000). The computed HI is then used for health monitoring and fault prognostic of bearings. Le Son et al. (2013) applied PCA in order to construct a HI named degradation indicator based on the first component of the PCA. The computed HI is used for RUL estimation of turbofan engine (Saxena et al. 2008b).




(a) Extracted features using WPD

(b) HI construction based on features fusion

Figure 2-6: HI construction example based on multiple features (Benkedjouh et al. 2013)

### Distance based HI

Some other works construct the HI by computing a distance between the evolving class (new incoming measures) and the normal class in a features space. Chammas et al. (2015) constructed a HI named severity indicator using two features, the computed HI represents how far the evolving class is from the normal class and how close it is getting to a failure class by using Kullback-Leibler (KL) divergence. The HI is then used for the prognosis of fault in a tank system (leak in the tank). In (Boškoski et al. 2015), a distance is computed between wavelet coefficient of fault free bearing and faulty bearing using Jensen-Rényi divergence, which measures the dissimilarities between probability distribution functions. In (Toubakh et al. 2015; Toubakh et al. 2016), two drift indicators are computed by the use of the Euclidean distance and

the Mahalanobis distance between normal and evolving classes. These two indicators are considered as HIs and applied for fault detection in the pitch system and the converter of a wind turbine.

#### Regression based HI

The goal of these techniques is to use regression models in order to map several features (input) into one HI (output). For example, a new HI for bearing monitoring is constructed using ANN in (Ali et al. 2015) in order to estimate the life percentage. The latter is a HI that shows the evolution of the system health state towards the failure, while the RMS, kurtosis, and RMS Entropy Estimator (RMSEE) are used as input of the ANN model. L. Guo et al. (2017) applied a Recurrent Neural Network (RNN) in order to merge six selected features and construct the HI, while the RNN are used to map the features to the HI which is between 0 and 1, where 0 represents the healthy state and 1 the faulty state of the component, the computed HI is employed for the RUL estimation of a test bed bearings and a wind turbines generator bearings.

### 2.4.3 Health indicator selection

In order to achieve fault prognosis and obtain an accurate RUL estimation, it is necessary to select the efficient HIs (features) to the evolution of degradation and discard the irrelevant ones. An efficient HI is the one that can follow the evolution of the degradation over time until failure, which means that it should be monotonically correlated with the degradation process. For this reason, selection criteria or metrics that can automatically select the most suitable HIs should be applied.

J. Coble et al. (2009) proposed three metrics in order to evaluate the HIs sensitive to the degradation evolution and select the best one. The proposed metrics are: "prognosability", "monotonicity", and "trendability". They are detailed in the next subsections.

### Prognosability

Prognosability, also called consistency, or failure consistency is a metric that returns a measure of the variance in the failure value of a population of degraded systems or components. A wide spread in failure threshold can make it difficult to accurately extrapolate a HI to the failure value. Prognosability measures the variability of the different HIs' values when the system is in failure. It is calculated as follows:

$$Prog = exp(-\frac{std(Vend)}{mean(|Vend - Vstart)|}) \tag{2.1}$$

where $Vend$ is the HIs' values when the system is in failure, while $Vstart$ denotes the HIs' values when the degradation starts.

### Monotonicity

The monotonicity evaluates the negative or positive trend of the HI, with the assumption that the system cannot self-heal. Monotonicity is measured by the absolute difference between the negative and positive derivative of HI, as indicated in the following equation:

$$Monot = \left| \frac{\text{Nb of } (d/dx > 0)}{n-1} - \frac{\text{Nb of } (d/dx < 0)}{n-1} \right| \tag{2.2}$$

where $d/dx$ represents the derivative of the HI, $n$ represents the number of observations, $Monot \in [0, 1]$, where 1 represents the perfect monotonicity.

### Trendability

Trendability is related to time and represents the correlation between the degradation trend and the operating time of a component, and it is calculated as follow (Javed et al. 2014)

$$Trend = \frac{|n(\sum_{i=1}^{n} x_i t_i) - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} t_i)|}{\sqrt{[n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2][n\sum_{i=1}^{n} t_i^2 - (\sum_{i=1}^{n} t_i)^2]}} \tag{2.3}$$

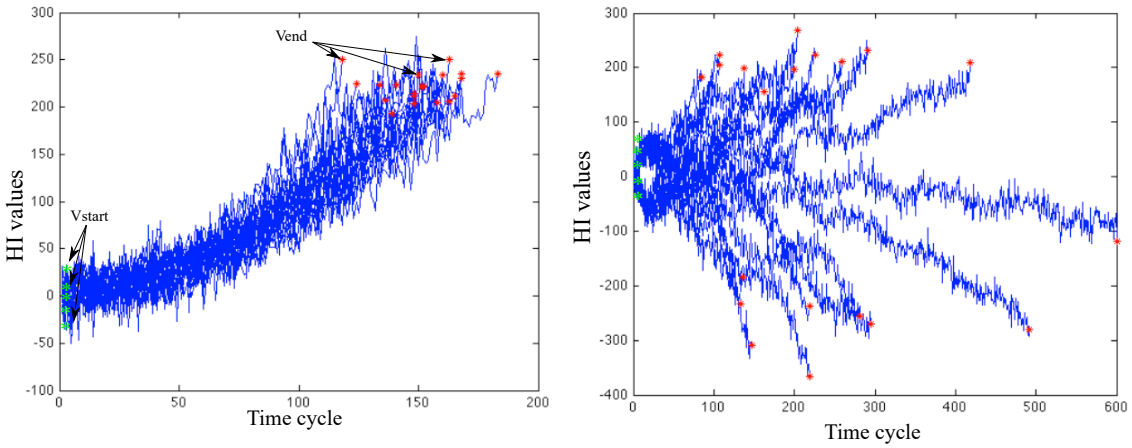
$Trend \in [0;1]$ represents the correlation coefficient between the value of HI for a pattern $x$ at time $t$. $Trend$ approaches 1 when the HI has a strong positive linear correlation with time.

### Robustness

B. Zhang et al. (2016) proposed the robustness metric in order to measure how robust the HI is to random fluctuations. It is computed as follow:

$$Rob = \frac{1}{n}\sum_{i=1}^{n} exp\left(-\left|\frac{xr_i}{x_i}\right|\right) \tag{2.4}$$

where $x$ is the HI values, and $xr$ denotes the random part values that is computed as the difference between the HI and the trend of the HI (smoothed HI).




(a) Population of good HIs

(b) Population of bad HIs

Figure 2-7: Example of good and bad population of HIs (J. B. Coble 2010b)

An example of a population of good and bad HIs is illustrated in Figure 2-7 (J. B. Coble 2010b). By comparing the population of good and bad HIs, shown in Figure 2-7a and Figure 2-7b respectively, it can be observed that the good HIs are characterized by their monotonic trend (monotonicity), correlation with time (trendability), clustered failure threshold (prognosability), and robustness.

Several works applied these metrics in order to select the most efficient HI for fault prognostic automatically. Saidi et al. (2017) computed 11 different time domain and

frequency domain HIs from vibration signals in order to estimate the RUL of a bearing in a shaft of a real wind turbine. The effectiveness of the HIs are compared using monotonicity and trendability metrics, and found that HI computed using spectral kurtosis (area under SK) presents better monotonicity and trenadability compared to other classical temporal HI, because it can avoid the effect of noise.

In (L. Guo et al. 2017; Abid et al. 2019), monotonicity and correlation (trendability) are used to select the suitable HIs among 14 extracted features. A linear combination between monotonicity and trendability is used in order to obtain a selection criterion as follows:

$$criterion1 = \frac{Monot + Trend}{2} \tag{2.5}$$

B. Zhang et al. (2016) computed the monotonicity, trendability, the robustness metric, then a weighted linear combination of the three metrics is applied in order to select the best HI as follow:

$$criterion2 = w_1.Monot + w_2.Trend + w_3.Rob \tag{2.6}$$

In (Duong et al. 2018), monotonicity, trendability, and robustness are applied in order to compare a new proposed HI based on discrete wavelet packet transform with two other HIs (RMS and variance). A weighted linear combination of the three metrics is applied in (Atamuradov et al. 2018) in order to rank several features. After that, the consistency metric (prognosability) is used to select the most consistent features among the ranked ones. L. Guo et al. (2018) proposed a new evaluation metric named scale similarity. It measures the scale similarity between the new HI and the HIs in the training set.

## 2.5 Degradation detection

Degradation detection is primordial for prognostic because it allows triggering the RUL estimation technique. Degradation detection aims to divide the health state of

a system into a healthy state (normal or nominal) and an unhealthy state (degraded or faulty). However, in some cases, the monitored system or component presents a gradually degraded trend during the whole operating period (e.g., flank wear of a milling tool (Eker et al. 2012)). Hence, there is no need to detect degradation since the degradation starts when the component is in operation.

The degradation detection is based on a HI (or HIs) that can follow the degradation evolution of the system. The simplest way to detect the degradation is to set an alarm threshold (Niu et al. 2010) on a single HI as illustrated in Figure 2-8.

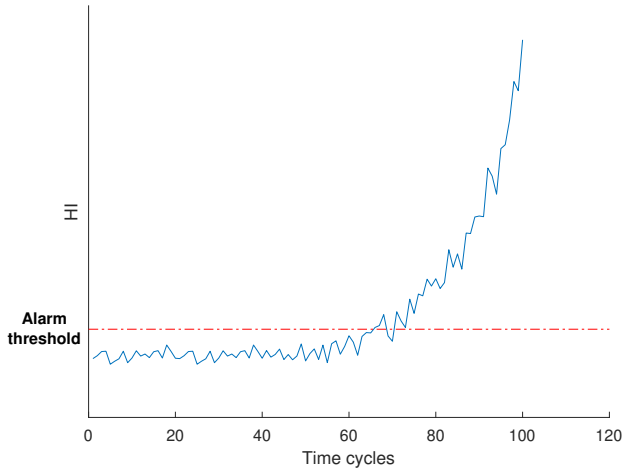



Figure 2-8: Degradation detection using alarm threshold

Another efficient widely used way for fault detection is to combine multiple HIs in order to exploit the complementarity of the HIs. This can be achieved using classification (machine learning) techniques such as support vector machine, k nearest neighbors, artificial neural network (ANN)...etc. The degradation can be detected using historical normal and faulty data, or by using only normal data by applying anomaly detection techniques. Figure 2-9 shows the classification of the degradation detection techniques.

### 2.5.1 Degradation detection using normal and faulty data

When using both normal and faulty (abnormal) data about the system, the goal is to find a boundary (separation) between the two classes (normal and faulty) as

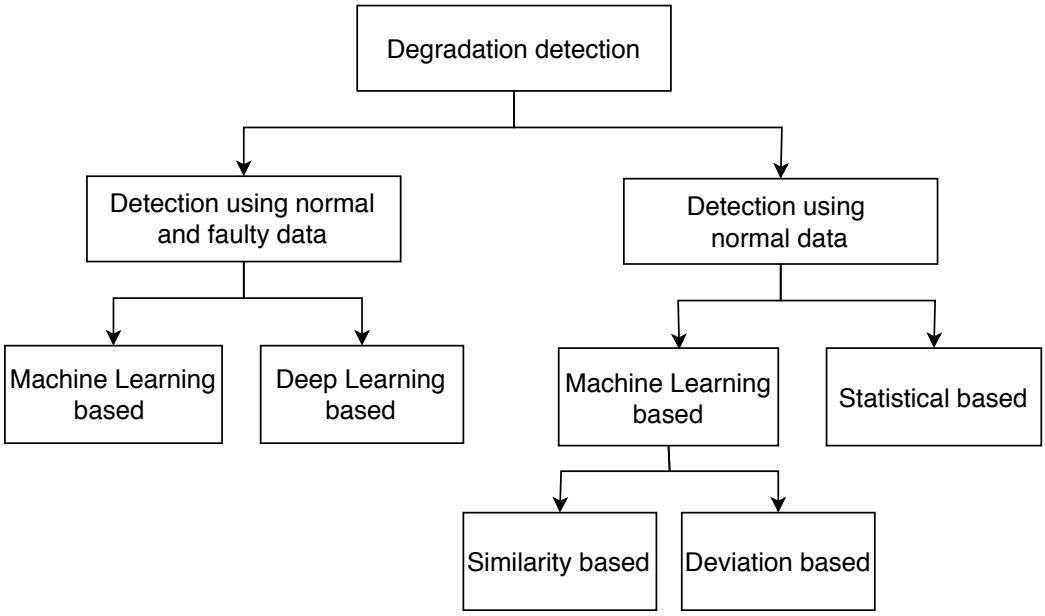



Figure 2-9: Classification of the degradation detection techniques

illustrated in Figure 2-10. Then, when a new measure is collected, it is assigned to one of those classes according to its similarity (closeness) to one of them. Degradation detection can be achieved using traditional machine learning models or by using recent deep learning models.

### Machine Learning techniques

K Nearest Neighbor (KNN) (Duda et al. 2012) is a classification method which measures the distance of the new observation from the K nearest points in the feature space. KNN is applied for fault detection and fault isolation in industrial systems (Z.-B. Zhu et al. 2011).

Support Vector Machine (SVM) is one of the most popular machine learning techniques for classification. The objective of SVM is to find the hyperplane in a specific space (i.e., transformed) separating two classes by maximizing the distance between both data sets (maximum margin) (Vapnik 2013). Soualhi et al. (2014) applied a multi class SVM by using a three dimensional feature space, in order to classify the health state of bearings into three states (class): good state, medium state, and degraded state. The medium state is a health state where it is difficult

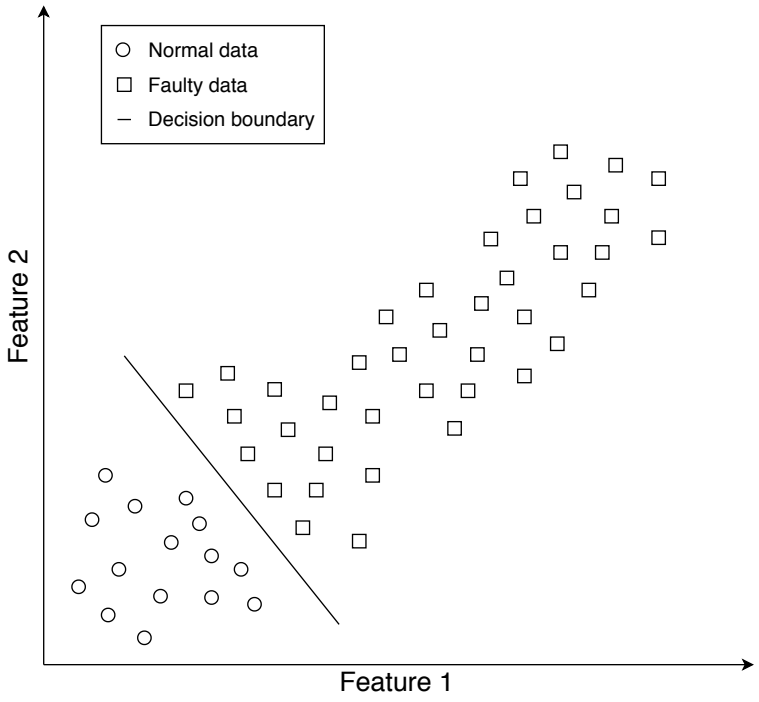



Figure 2-10: Classification using normal and faulty data

to differentiate between the normal and degraded state of the component, where in the degraded state the degradation or fault occurrence is confirmed. In (Saidi et al. 2015), SVM is applied for fault diagnosis of bearings in order to distinguish four kinds of bearings fault: healthy bearing, inner race fault, outer race fault, and ball fault. Laouti et al. (2011) proposed a fault detection and isolation approach based on SVM, which is validated using a simulated wind turbine benchmark (Odgaard et al. 2009). Artificial Neural Networks (ANNs) are more and more used for fault diagnosis. Saxena et al. (2007) applied ANN for fault classification of rotating mechanical system (bearings), where a genetic algorithm is employed for selecting the optimal set of features in order to achieve better classification accuracy.

### Deep Learning techniques

Recently deep learning techniques are used for fault diagnosis. They are a deep representation of the ANN. Their advantages are that there is no need to achieve health indicators extraction, the diagnosis is achieved using only raw collected data. However, these techniques require an important size of faulty data for training containing

information about all the faulty conditions. W. Yan et al. (2019) proposed a deep learning based approach for fault detection by combining Autoencoder for feature extraction, then the features are fed to an Extreme Learning Machine (ELM) which is a particular type of ANN (Huang et al. 2006). The proposed approach is applied for fault detection in gas turbine combustors.

Convolutional Neural Network (CNN) (LeCun et al. 1998) has been proven successful in many domains. The main advantage of CNN is the ability to extract sensitive features to the degradation automatically without manual feature extraction and selection. Janssens et al. (2016) compared CNN with a classical manual feature extraction approach for bearings fault detection. The obtained results using automatic features extraction based on CNN showed better accuracy than the detection using manual feature extraction approaches. Hierarchical CNN is applied in (X. Guo et al. 2016) for bearings fault diagnosis, where CNN detects the fault occurrence and isolates the element responsible of the fault (outer race, inner race, and ball bearing).

Long Short Term Memory (LSTM) is a deep learning technique that can deal with sequential data. It is a Recurrent Neural Network (RNN) that can model long-term dependencies hidden in sequential data (Hochreiter et al. 1997). LSTM is used in (J. Lei et al. 2019) for condition monitoring and fault diagnosis of a wind turbine. It showed good performance compared to other techniques such as SVM, Multi Layer Perceptron (MLP), simple RNN, and convolutional neural network.

### 2.5.2 Degradation detection using normal data

The degradation detection using normal data can be achieved using anomaly detection (novelty detection, outlier detection) also called one class classification, which refers to the problem of finding patterns in data that do not conform to the expected behavior (Marsland 2003). The objective of anomaly detection methods is to detect degradation and fault using only the data collected from the system or its components under normal (nominal) operation conditions. These techniques can be applied to degradation detection since the data points about normal operating conditions are well condensed in the feature space, while the data points about degradation operation

conditions are spread in the feature space as illustrated in Figure 2-11. Degradation detection using normal data aims at finding a boundary around the normal data as represented in Figure 2-11, then the new measurement is classified as normal or abnormal if it is inside or outside the boundary, respectively.

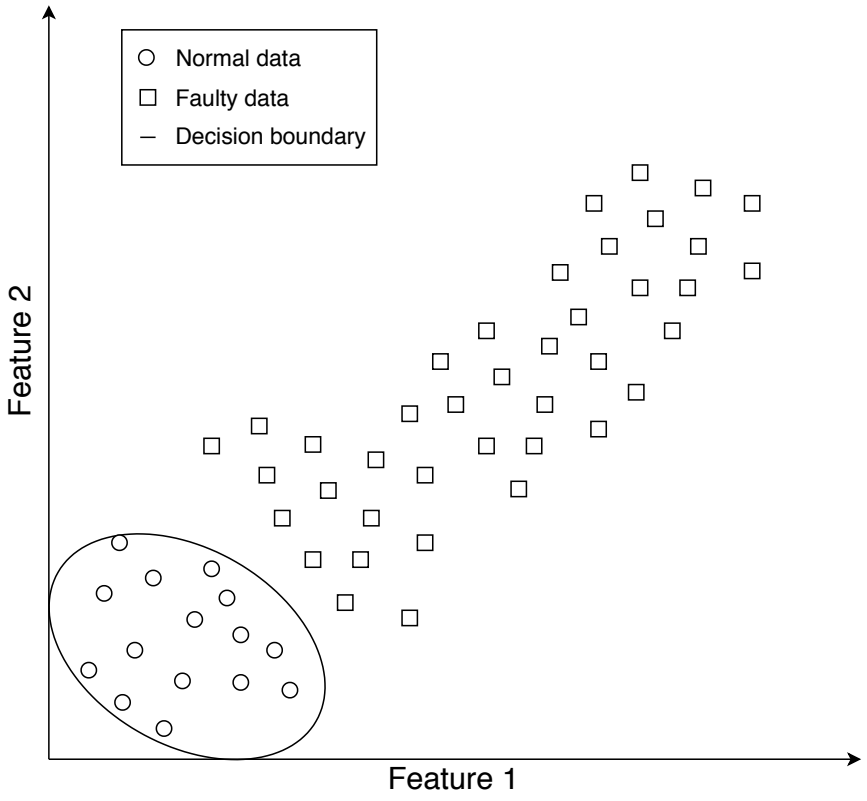



Figure 2-11: Classification using only normal data

Anomaly detection methods can be broadly categorized into statistical, or machine learning techniques that include similarity based, and deviation based methods (An et al. 2015).

### Statistical techniques

Statistical anomaly detection techniques assume that the data is modeled from a specific probability distribution. The simplest way for anomaly detection is to represent the normal data as a Gaussian distribution, then apply $3\sigma$ interval ($\mu + 3\sigma$ contains 99.7% of data instance) in order to set the threshold and detect the anomaly (Chan-

dola et al. 2009). Jin et al. (2016) applied the box-cox transformation in order to change the distribution of the HI into Gaussian distribution, then applied an alarm threshold for fault detection in order to trigger the RUL estimation step for bearings faults.

### Machine Learning techniques

Similarity based techniques compute the similarity (closeness) between the new measurement point and the normal class in the feature space. The similarity can be computed using a distance metric such as Euclidean, Mahalanobis, and Manhattan distance. Y. Wang et al. (2016) computed Mahalanobis distance between the new measurement and the original healthy state, then $3\sigma$ interval is set as a threshold in order to detect the fault. KNN is also a similarity based method, it is adapted in (He et al. 2007) by examining the distance of a new sample to its neighboring training samples (using only normal data) for fault detection in an industrial process (semiconductor manufacturing process).

SVM can also be a similarity based method because it computes the distance of the new observation from the support vector points. Schölkopf et al. (2000) developed a variant of SVM in order to tackle the one class classification problem called One Class Support Vector Machine (OCSVM). The goal of OCSVM is anomaly detection and novelty detection, which is extensively applied in the literature (Khan et al. 2014). OCSM is also applied in (Yin et al. 2014) for fault detection. Shin et al. (2005) applied OCSVM for machine fault detection, where the results showed that the OCSVM performance is comparable to MLP neural network performance. An approach based on OCSVM is proposed in (Fernandez-Francos et al. 2013) for fault detection and isolation of bearings. The proposed approach showed its ability to detect failures in an incipient stage, to isolate its location, and qualitatively assess its evolution over time.

Deviation based methods use the reconstruction errors as the anomaly score (which can be used also as HI). Autoencoder is a type of neural network that can reconstruct the input at the output layer (Hinton et al. 1994; Bengio et al. 2013). It

has been used in the literature to extract features for fault detection in an unsupervised manner (Sun et al. 2016; Haidong et al. 2018). Anomaly or novelty detection can be achieved when the reconstruction errors (residual) between output and input is greater than a threshold. Jiang et al. (2017) applied a variant of the autoencoder named denoising autoencoder for errors generation, while a threshold is used for fault detection. The proposed approach is validated on a wind turbine benchmark and then on a real wind turbine data. The construction of the residual based HI is also a deviation based methods since the residual based HI is the deviation between nominal operation conditions and the new measurement. Then, the degradation detection can be triggered by setting an alarm threshold, while the latter is generally set as the maximum value of residual under normal operation.

## 2.6 Remaining useful life estimation approaches

The degradation detection or diagnosis (detection and isolation) triggers the estimation of the RUL. The RUL is the time between the failure time when the extrapolated HI exceeds the failure threshold and the present time. The RUL is computed since the degradation is detected (alarm threshold reached). The RUL estimation process is illustrated in Figure 2-12.

Several categories of prognostic approaches were applied in the literature for achieving prognostics (RUL estimation), each category is efficient in a particular situation depending on the availability of data, availability of the physical model, and availability of information about the current health state. Most of the review works in the literature classify the prognostic approaches into model based (physics based) and data-driven based approaches (Goh et al. 2006; Heng et al. 2009). A taxonomy of model based, data-driven based, and hybrid approaches are proposed in (Lee et al. 2006), where the hybrid approaches combine the first two approaches. Tobon-Mejia et al. (2012) classified the prognostic approaches into model-based, data-driven based, and experience based prognostic approaches, where the latter use the traditional reliability methods.

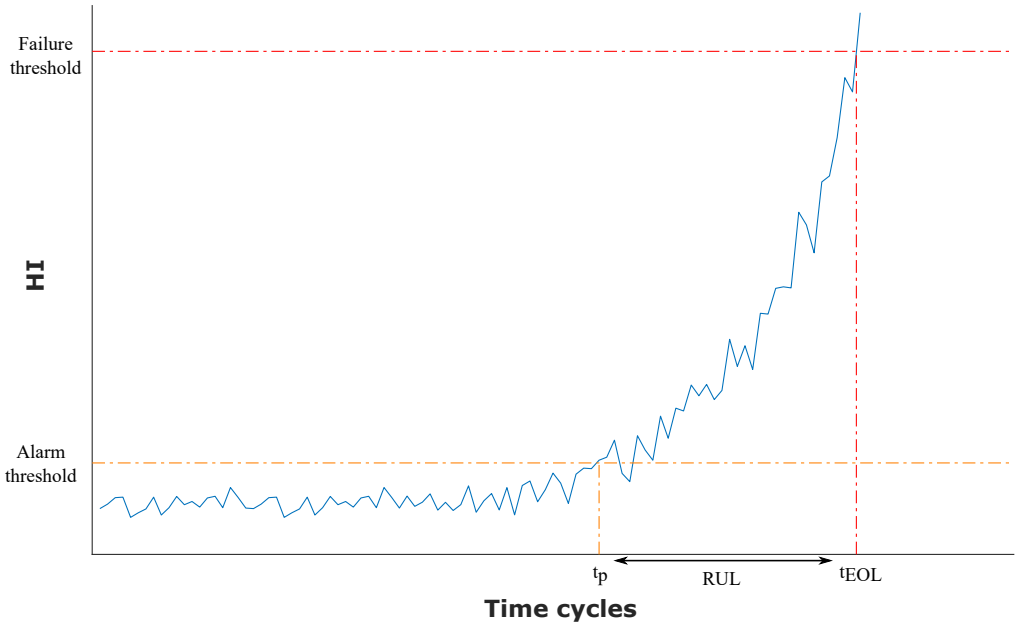



Figure 2-12: Illustration of RUL estimation

In this manuscript, the RUL estimation is classified into two main approaches: Experience based approaches and degradation modeling approaches (see Figure 2-13). Experience based prognostic is achieved by applying reliability or similarity based approaches. The degradation modeling can be achieved by using physical models or by data-driven approaches (Abid et al. 2018).

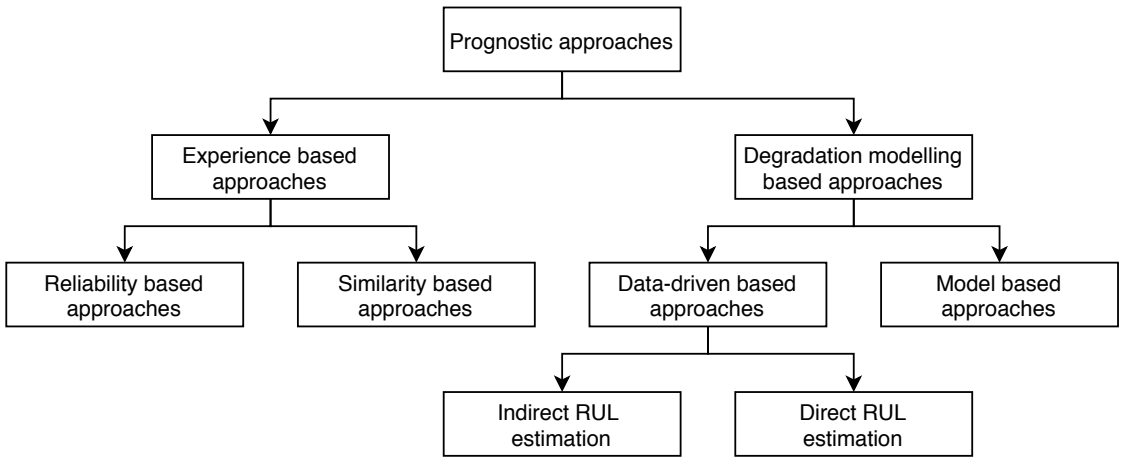



Figure 2-13: Classification of RUL estimation approaches

Experience based approaches are based on the requirements of multiple run-to-

failure experiments in order to achieve prognostics. The run-to-failure experiment or sequence represents the operation of the system or component during the whole degradation process until failure. These sequences can be collected during the operation of the component, but in general, they are generated artificially in the laboratory by accelerating the degradation process. Experience based approaches can be classified into reliability based approaches and similarity based approaches.

### 2.6.1 Reliability based approaches

Traditional reliability based approaches use several run-to-failure sequences in order to characterize the expected lifetime of a population of components (systems). These approaches can be applied even when no information is available about the current health state of a component. Besides, the operating conditions are not considered for lifetime estimation. Usually, these approaches are applied in the manufacturing industry in order to estimate the mean life of a product, which is determined by analyzing time-to-failure of product population with similar characteristics. For this analysis, a failure distribution is applied such as Weibull distribution which is widely applied for characterizing the time-to-failure probability function of the component (product) (Schömig et al. 2003; Zhai et al. 2013).

### 2.6.2 Similarity based approaches

Similarity based approaches also use a huge amount of a priori run-to-failure sequences (library of degradation trajectories or patterns) for a set of components under different operation conditions. This category of approaches requires the monitoring of the current health state or health indicator of the system (component). Then the current HI sequence is compared with the library of degradation trajectories. The goal is to select the degradation trajectory (degradation evolution) that best matches, in terms of similarity or closeness, the degradation evolution of the current component. The selected degradation evolution is used to predict the RUL of the current component. T. Wang et al. (2008) employed an Euclidean distance in order to find the most similar

HIs with the current one. Then, the best HIs are used for RULs estimation while the final RUL is computed using a weighted mean of the computed RULs. The proposed approach is validated using run-to-failure data collected from a faulty aircraft engine datasets.

Some other works apply KNN for finding the similar HIs from the library. The selected HIs are represented as the K nearest neighbors according to defined distance metric (Ramasso et al. 2012; Mosallam et al. 2016). Mosallam et al. (2016) used KNN in order to find the most similar trajectories in training to the online one, while a recursive discrete Bayesian filter is applied for RUL estimation. The proposed method is evaluated using two data sets, namely, turbofan engines and lithium-ion battery (Saxena et al. 2008a; Saha et al. 2007). Zio et al. (2010) applied a fuzzy pointwise similarity concept in order to match the current data to the data in the library of reference patterns. Then the RULs of the selected reference patterns are aggregated using a weighted mean for the final RUL estimation.

Degradation modeling based approaches aim at modeling the degradation dynamic evolution and predicting its progress over time until failure. Precisely, those approaches aim at estimating the RUL considered as the time between the starting time of the degradation detection and the time of end of life. The degradation dynamic evolution can be modeled using model based approaches or data-driven based approaches. This thesis focuses on data-driven approaches.

### 2.6.3 Model based approaches

Model based approaches require specific physical knowledge about the system operation and fault evolution and development. They use physical and mathematical laws and principals in order to model the degradation trend. Paris law (Paris et al. 1963) is widely used for modeling damage propagation in a material. Paris law is a crack growth equation that gives the rate of growth of a fatigue crack. It is applied in (Kacprzynski et al. 2004) for the fault prognosis of gears. Oppenheimer et al. (2002) proposed a physics based approach for performing fault diagnosis and prognosis. The observer method is used for fault detection and diagnosis, where a life model based on

material crack growth laws is used for the RUL estimation. In (Y. Lei et al. 2016b), a particle filter based method is proposed to predict the RUL of machinery with degradation processes described using a variant of Paris model, the proposed method is validated using accelerated degradation test of bearings.

### 2.6.4 Data-driven based approaches

Data-driven based approaches require few or several degradation data (depending on the used method) in order to estimate the RUL. They build a model that learns using only the available data. These models can be statistical or based on Artificial Intelligence (AI) methods. First, the model is fitted with the historical degradation data offline, then when the degradation is detected the model permits to estimate the RUL online. RUL estimation using data-driven approaches can be classified into indirect and direct RUL estimation.

#### Indirect RUL estimation

The indirect RUL estimation is the standard RUL estimation where HIs are defined using features extracted from the new incoming data. Then, the trend of the HIs is extrapolated using a data-driven model (statistical or AI), where the RUL is deduced as remaining time to the failure (see Figure 2-14).

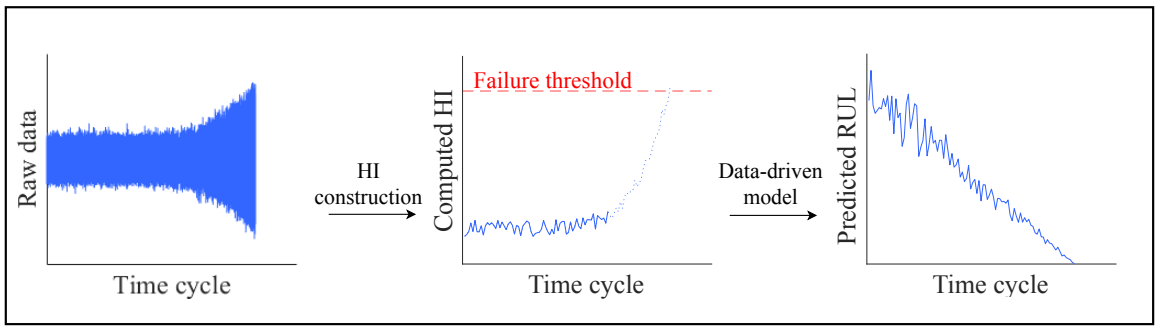



Figure 2-14: Indirect RUL estimation

Statistical based methods for prognostic use build a model based on empirical knowledge without having physical knowledge about the system operation. Some of these methods could have an easy update of their parameters because the model

contains only a few parameters to fit. Hence, they can perform prognosis with few or without using any a priori degradation sequences for RUL estimation. Qian et al. (2014) utilized the Auto Regressive (AR) model for RUL prediction of bearings. The AR model assumes that future degradation evolution is dependent on past observations.

Two Moving Average (MA) windows are used in (Dupuis 2010) for RUL estimation of bearings in a wind turbine gearbox by monitoring the oil debris signal. Auto Regressive Moving Average (ARMA) is a traditional statistical model for time series prediction, which combines the AR model and the moving average model. It is applied in (J. Yan et al. 2004) for RUL estimation of an elevator door motion system. Ahmad et al. (2017) developed an adaptive predictive method for RUL estimation based on the indicator RMS and the quadratic regression model. The model parameters are updated with each new time cycle, hence, it does not require several a priori degradation sequences for RUL estimation. The developed method is evaluated using bearings degradation data. The exponential model is also widely used to characterize the degradation evolution of bearings (Shao et al. 2000; Gebraeel 2006). Gebraeel (2006) developed a sensory updating method using the exponential model where the model is updated for each new incoming sample from the sensors. A double exponential model is applied in (Jin et al. 2016) which is the summation of two exponential models, whereas the model parameters are updated using kalman filter. The developed model is used for modeling the degraded trend of bearings. Also, a double exponential model is applied in (L. Guo et al. 2017) in order to extrapolate the computed HI of degraded bearings, the approach is validated using experimental data of bearings testbed and real data from a generator bearing of a wind turbine.

Artificial intelligence methods attempt to learn the degradation evolution of the system, by training a model on the available a priori sequences about the degradation. They are able to perform prognostics for complex dynamic systems. In that case, the degradation is difficult to be characterized by model based approach or statistical approach. However, compared to the statistical methods, AI methods require an important number of run-to-failure data in offline for training the model. Support

Vector Regression (SVR) is a variant of SVM for regression, it is an AI method extensively used for RUL estimation in the literature. Saidi et al. (2017) applied SVR in an indirect way for predicting the trend of the HI and estimate the RUL of a high speed shaft bearing of a wind turbine. In (Soualhi et al. 2014), an approach based on three SVR models is developed, it is used for predicting the evolution of three computed HIs on vibration signal collected for health monitoring of bearing, then the smallest RUL is deduced as the final predicted RUL. SVR is also used in (Benkedjouh et al. 2013) for RUL estimation of bearings by using the indirect way.

ANNs are the most commonly AI techniques used for RUL estimation. They can model the complex non-linear relationship between input and output. An indirect life percentage estimation method is developed in (Ali et al. 2015), where the life percentage is the inverse of the RUL. The life percentage of bearings is estimated by predicting the HI evolution based on ANN. An indirect RUL estimation is applied in (Malhi et al. 2011), while the trend evolution of the computed HI is predicted using RNN. LSTM is a RNN employed to learn the long term dependencies, its architecture makes it able to remember information for long periods of time. Y. Zhang et al. (2018) applied LSTM in the indirect way by modeling the evolution of the capacity degradation trajectories for the RUL estimation of lithium-ion batteries.

### Direct RUL estimation

In direct RUL estimation, the RUL is estimated directly without trend prediction until failure, the input of the AI model can be either raw data or extracted features while the output is the actual RUL (see Figure 2-15). However, these methods are black box models, hence, it is difficult to have interpretable elements to share with operators.

Different artificial intelligence techniques are applied in the literature, especially machine learning methods, and they are used to model a nonlinear mapping between the output and the input space. Hidden Markov Model (HMM) is also used for RUL estimation thanks to their advantage of dealing with data sequentiality. It is applied in (Tobon-Mejia et al. 2012) combined with a mixture of Gaussians for predicting the

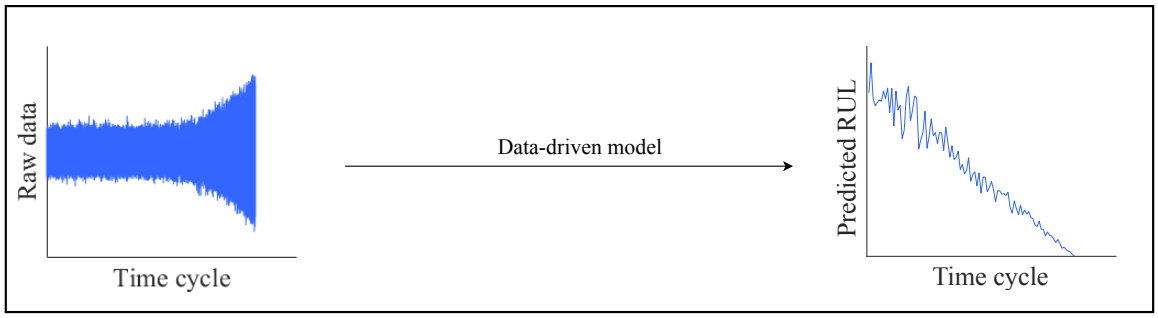



Figure 2-15: Direct RUL estimation

failures in bearings. W. Q. Wang et al. (2004) developed an RUL estimation method based on a neuro-fuzzy system, which is a neural network-based fuzzy system. The structure of the fuzzy inference is determined by expertise, where its membership functions are trained by using ANN. In (Aye et al. 2017), the remaining life is predicted for slow speed bearings based on acoustic emission thanks to the Gaussian Process Regression model (GPR), GPR is a flexible non-parametric Bayesian model that permits a prior probability distribution to be defined over functions directly.

Loutas et al. (2013) used the SVR in a direct way where the inputs are the selected HIs and the output is the True RUL, it is applied for RUL estimation of rolling bearings. Z. Tian (2012) developed an ANN based method for RUL estimation of equipment using as input the age and conditions monitoring values at the present and past measurement. The developed method is validated using real world vibration monitoring data collected from pump bearings. In (Mahamad et al. 2010), a feed forward neural network is applied for direct prediction of the bearing life percentage. Recurrent Neural Networks (RNN) are as well used for RUL estimation, they contain an internal memory and can learn complex nonlinear mapping. Heimes (2008) proposed a direct RUL estimation for turbofan engine based on RNN, where the proposed method based on RNN yields better results compared to ANN.

Recently deeper architectures of the ANN are applied for RUL estimation named deep learning models. Deep learning models are more and more used for machine health prognostics and have proven their effectiveness for RUL estimation. Most of the deep learning based approach for prognostics estimate the RUL in a direct

way. As explained before, the advantage of these techniques is that there is no need to achieve health indicators extraction, and they require several a priori sequences about the degradation evolution for RUL prediction.

As for fault diagnosis, the most applied methods for fault prognostics are CNN and LSTM. CNN and LSTM have shown their effectiveness for RUL estimation applied on different applications (e.g., bearings, battery, and aircraft engine). CNN has a deep architecture which makes it suitable for achieving automatic features extraction without computing the features manually. Babu et al. (2016) applied CNN for direct RUL estimation of data generated from a turbofan engine. The raw data collected from different sensors are used as input, whereas the true (actual) RUL is used as output.

Deep CNN is applied in (X. Li et al. 2018) for RUL estimation using the turbofan engine dataset, the proposed architecture is deep because it stacks five convolution layers in order to capture the representative information from raw input data. It is applied in (X. Li et al. 2019) for direct RUL estimation of bearings by using as input a time-frequency transformation of the raw vibration signal and the actual RUL as output. LSTM is also widely used for the direct RUL estimation. Several works applied the LSTM in order to predict the RUL of an aircraft engines (S. Zheng et al. 2017; Hsu et al. 2018). Mao et al. (2018) applied the LSTM in a direct way by using as input features extracted from the vibration signal in order to predict the RUL of bearings. Some works also used a variant of the LSTM named Bidirectional LSTM (BLSTM) that can learn the dependencies of sensor data in both forward and backward direction. BLSTM is applied for the RUL estimation of an aircraft engine in (J. Wang et al. 2018; J. Zhang et al. 2018).

## 2.7 Remaining useful life evaluation metrics

The predicted RUL is evaluated in a post prognostic step after failure occurrence using suitable and meaningful metrics. When the system's failure is reached, the actual or true RUL can be computed. This true RUL is then used as a reference to find the

error between it and the predicted RUL. The true RUL is inversely proportional to the degradation rate and is computed as the difference between the present time and failure time, as illustrated in Figure 2-16.

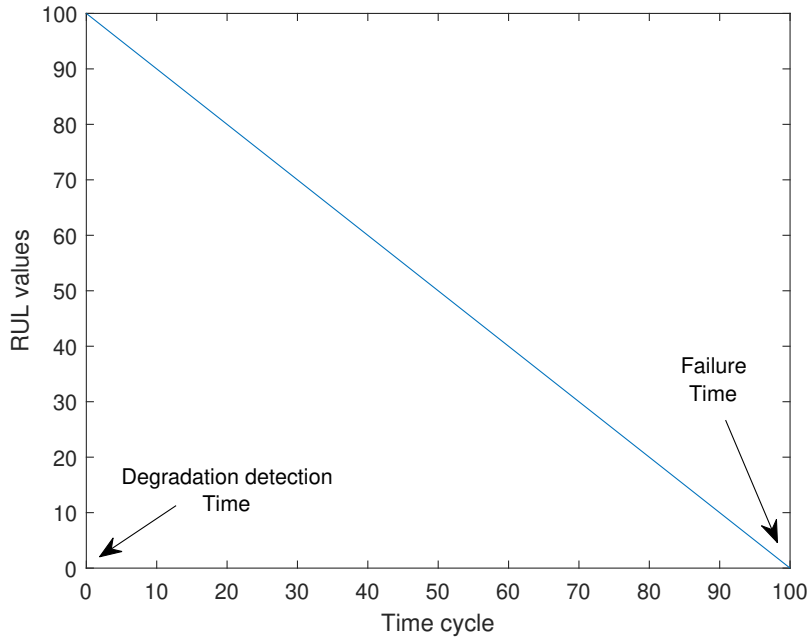



Figure 2-16: True RUL vs time

In order to quantify the performance of the computed RUL, errors based metrics can be computed between the predicted and the true RUL. Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) are used for RUL evaluation (Tobon-Mejia et al. 2012; Loutas et al. 2013). RMSE is the standard deviation of the prediction errors, where MAPE is a relative error that can express the accuracy as a percentage:

$$RMSE = \sqrt{\frac{1}{n_t}\sum_{t=1}^{n_t}(true\_rul(t) - rul(t))^2} \tag{2.7}$$

$$MAPE = \frac{1}{n_t}\sum_{t=1}^{n_t}\left|\frac{true\_rul(t) - rul(t)}{true\_rul(t)}\right| \tag{2.8}$$

where $n_t$ is the number of time index from degradation detection until failure, $t$ is the time index (time cycle), $true\_rul$ represents the actual RUL, and $rul$ represents

the predicted RUL.

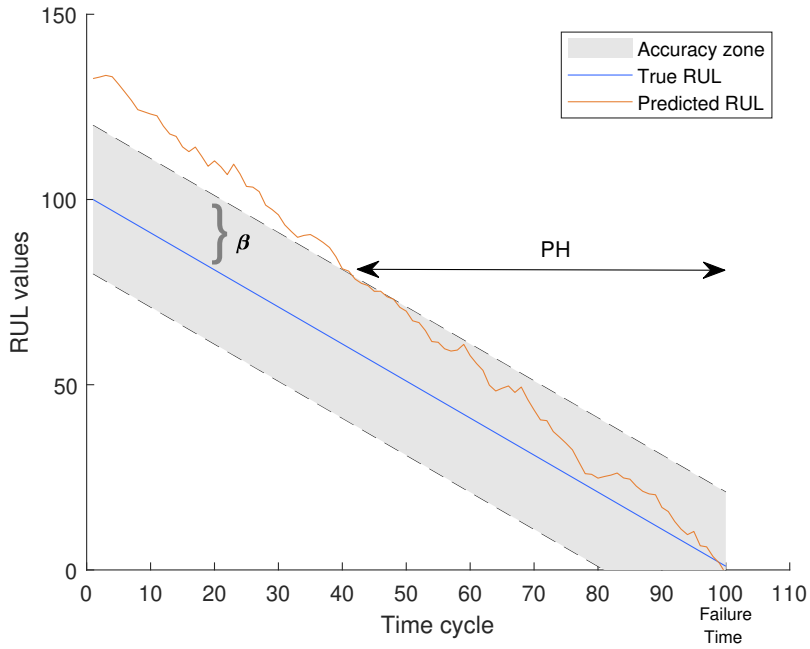



Figure 2-17: Prognostic horizon

Saxena et al. (2008c) proposed several metrics for RUL evaluation that are widely used in the literature. The most relevant metrics are: prognostic horizon, $\alpha - \lambda$ performance, relative accuracy, and cumulative relative accuracy. Prognostic Horizon (PH) is the difference between the time of the end of life ($t_{EOL}$) and the present time when the RUL is lower than a certain threshold defined by parameter $\beta$ (see Figure 2-17).

$\alpha - \lambda$ performance can determine whether the prediction falls within specified limits $\alpha$*100% at particular distance $\lambda$. For example, in Figure 2-18 the RUL prediction falls within the cone +/- 30% (i.e., $\alpha$=0.3) at a halfway distance ($\lambda = 0.5$) from degradation detection ($\lambda = 0$) until failure ($\lambda = 1$).

Relative Accuracy (RA) is the error between the predicted RUL relative to the actual RUL (true RUL) at a specific time index $t$.

$$RA(t) = 1 - \frac{|rul(t) - true\_rul(t)|}{true\_rul(t)} \tag{2.9}$$

Cumulative Relative Accuracy (CRA) $\in [0, 1]$ is the weighted average of the Rel-

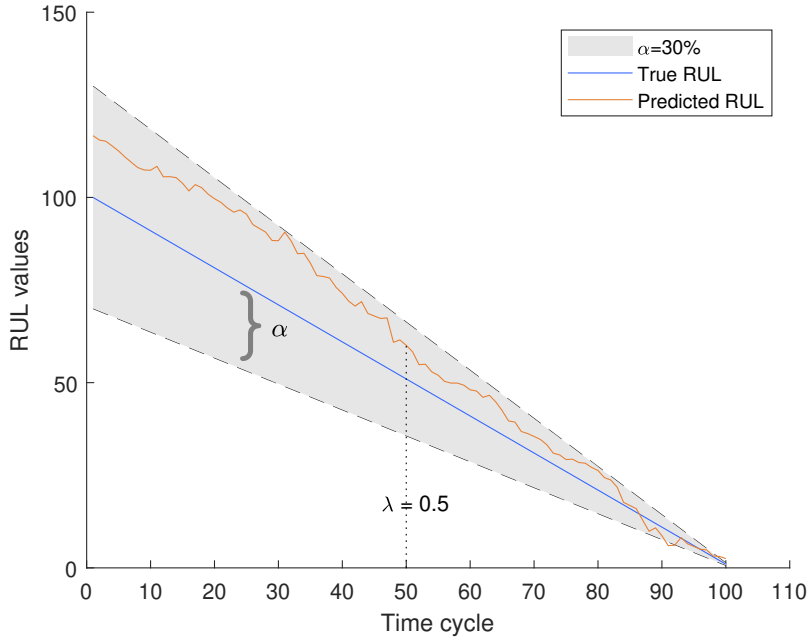



Figure 2-18: $\alpha - \lambda$ performance metric

ative Accuracy (RA). Compared to RA that evaluates the RUL at a specific time instance, CRA can evaluate the predicted RUL at multiple time instances, and it has been widely used for the RUL evaluation (Ahmad et al. 2017; Tobon-Mejia et al. 2012; Duong et al. 2018). CRA is calculated as follows:

$$CRA = \frac{1}{n_t} \sum_{t=1}^{n_t} w(t) RA(t) \tag{2.10}$$

$$w(t) = \frac{\frac{1}{true\_rul(t)}}{\sum_{t=1}^{n_t} \frac{1}{true\_rul(t)}} \tag{2.11}$$

where $n_t$ is the number of time index (time cycle) from RUL estimation start until failure (End Of Life (EOL)), and $w(t)$ is a weight factor as a function of the actual RUL (Goebel et al. 2011). The weights aim to improve the RUL evaluation by penalizing more the errors closer to $t_{EOL}$.

## 2.8 Discussion

Achieving prognostics requires to develop an approach with several steps. The main steps are HI construction, degradation detection (or diagnosis), and then modeling the degradation evolution for the RUL estimation step.

Health indicator construction is the first and main step for achieving prognostics. It is preferable to have some knowledge about the system (expert knowledge) in order to choose an efficient HI. Alternatively, the HI can be selected automatically among a library of condition-based indicators using some evaluation metrics such as prognosability, monotonicity, trendability, and robustness.

For the degradation detection, it is efficient to apply detection using normal and faulty data (i.e., classification techniques) by exploiting multiple health indicators in order to find the optimal boundary between normal and faulty data points. However, in real industrial systems, there is a lack of historical degradation data and sometimes they are not available (e.g., for new systems). In the latter case, it is preferable to apply anomaly detection methods because the historical data about normal operation conditions are widely available.

The discussed prognostic approaches in this manuscript are compared according to their potential requirements in Table 2.1, where each requirement can be "Required", "Not required", and "Beneficial" for each prognostics approach.

Table 2.1: Prognostic approaches requirements

| RUL estimation Approaches | | Physical model | Degradation history | Current health state | Degradation detection |
|---|---|---|---|---|---|
| Experience based approach | Reliability | Not required | Required | Not required | Not required |
| | Similarity | Not required | Required | Required | Required |
| Degradation modeling bsed approach | Model based | Required | Beneficial | Required | Required |
| | Data-driven | Not required | Beneficial | Required | Required |

Experience based approaches (reliability and similarity) do not require an engineering model, but they require several run-to-failure sequences to achieve RUL estimation. Also, reliability based approaches do not require current health state and degradation detection comparing to other approaches. Degradation history is not required for Model based approaches, because they require physics or mathematical

model about the system operation and degradation evolution. Data-driven based approaches do not require an engineering model about the system, and it is beneficial to have degradation history. They can deal with a huge amount of run-to-failure history (i.e., using deep learning methods), and deal with the lack of degradation history by using statistical methods (i.e., HI-extrapolation methods). In addition, the latter can be adaptive to the current degradation evolution.

Table 2.2 presents a comparison between the four prognostics approaches in terms of precision, implementation, cost, and interpretability. Precision criterion denotes the RUL estimation's accuracy provided by the different approaches on component level and system level. Implementation refers to the ease of implementation of the approach, where the cost criterion is the cost of implementing the methods and generating degradation data in order to achieve the prognostics. Finally, interpretability is the possibility of explaining how the RUL is estimated. The sign (+) refers to the advantage and (-) refers to the drawback of the methods.

Table 2.2: Prognostic approaches comparison

| RUL estimation Approaches | | Precision | | Implementation | Cost | Interpretability |
|---|---|---|---|---|---|---|
| | | Component level | System level | | | |
| Experience based approach | Reliability | + | - | ++ | - | - |
| | Similarity | + | - | ++ | - | - |
| Degradation modeling bsed approach | Model based | ++ | + | - | - | + |
| | **Data-driven** | + | ++ | ++ | + | − |

Experience based approaches (reliability and similarity) are easier to apply for RUL estimation since the online HI is compared with the offline HI trajectories. However, they require a huge amount of historical run-to-failure sequences (high cost). Model based approaches may have a good precision at the component level (e.g., crack propagation of bearing). However, when the system is more complex, this kind of approaches may not be applicable since it is challenging to build a physical model where several components are in interaction. Despite the lack of interpretability of the data-driven approaches, they are the most suitable (best trade-off) to perform the prognostic task when comparing with other approaches in terms of precision, implementation, and cost.

For developing a data-driven approach that can achieve the prognostics, the avail-

ability of historical degradation data is the main requirement that should be considered. When there is a lack of historical run-to-failure data, adaptive techniques should be investigated, such as HI-extrapolation methods (i.g., linear regression, quadratic regression, exponential model...). The advantage of such techniques is the non requirement of several run-to-failure data, they are applied in the indirect RUL estimation way by computing the HI, and then extrapolate it until reaching the failure threshold. The failure threshold can be set by some failure data if available or set by expert knowledge otherwise.

When several run-to-failure data are available in offline, direct estimation using more sophisticated techniques is preferable, such as machine learning or deep learning techniques (i.g., HMM, SVM, ANN, LSTM, CNN). Despite the requirement of several run-to-failure data, these methods possess several advantages, they can estimate the RUL directly without modeling the evolution of a HI, and raw data can be used as input without computing health indicators when applying deep learning techniques. Moreover, they can model a complex non-linear relationship between the input and the output (RUL). Hence, they can be applied to estimate the RUL for complex dynamic systems switching between different operating conditions in variable environment conditions (e.g. wind turbine). Furthermore, the failure threshold can be set only the training set.

## 2.9 Conclusion

This chapter presented the three main steps of the prognostics and health management strategy: health indicator construction (data processing), degradation detection (diagnostics), and RUL estimation (prognostics). The HI can be based on single or multiple indicators, the use of one HI is better for keeping the interpretation of the HI, whereas fusing multiple HI may result in a virtual HI without interpretation that could be more efficient for the degradation detection and RUL estimation. The degradation can be detected using both data of the system under normal and faulty conditions. However, the normal data are widely available in real industrial systems

where there is a lack of faulty data. In this case, it is preferable to apply anomaly detection methods while using only normal data as a priori. When the incipient fault is detected, the prognostic approach is triggered. The RUL can be estimated using experience based approaches (reliability and similarity), model based approaches, and data-driven based approaches. Data-driven approaches appear to be the most suitable in terms of precision, implementation, and cost. Data-driven approaches can be efficient in both cases when having few or having a huge amount of historical degradation data.

In the next chapter, a data-driven approach is proposed for the fault prognostics, which deals with insufficient historical degradation data (without and with few historical degradation sequences). The RUL is estimated using the indirect RUL estimation since these methods can be adaptive and do not require several a priori sequences for RUL prediction. The approach will be validated using different experimental datasets (high speed shaft of a wind turbine, rolling bearings, and aircraft engine).

# Chapter 3

# Prognostic approach with insufficient a priori degradation sequences

## 3.1 Introduction

In industrial systems, a large amount of historical data about the normal (healthy) operating conditions are often available, while historical data about degraded and faulty conditions are often unavailable because of their high cost, or for safety reasons, or in the case of newly installed machines. Besides the high cost of generating degradation data in laboratory conditions, the system's degradation behavior is often different in real operation conditions due mainly to the variation of environmental and load conditions.

In order to overcome these aforementioned limits, this chapter proposes a data-driven prognostic approach dealing with the problem of insufficient historical run-to-failure sequences. It performs prognostics when there is no run-to-failure sequences, or only a few available ones, insufficient to allow a reliable and precise RUL estimation. The proposed approach comprises two main steps: degradation detection and localization and RUL estimation. In the first step, the degradation is detected using only data collected when the system is working under normal conditions. The second step estimates the RUL using blind or informed path. The blind path is triggered when no a priori sequences are available offline, where the best HI is selected dynamically

by using a selection criterion. Then, the RUL is predicted using an adaptive model, which is updated with each new incoming point. The informed path is triggered to estimate the RUL when few a priori sequences about the degradation evolution are collected. In this path, the best HIs as well as the best models are selected according to their capacity to perform a precise estimation of RUL over time on the few a priori sequences. Hence, several HI-model combinations (pairs) are used for the RUL prediction in order to improve the accuracy as well as the reliability of RUL's estimation.

The chapter is organized as follows. Section 3.2 presents the proposed approach for RUL estimation with insufficient a priori sequences. The proposed approach is validated in sections 3.3, 3.4, and 3.5 using, respectively, vibration data collected from a real degraded bearing of a wind turbine high speed shaft, and vibration data generated from a degraded bearing with inner and outer race fault. Also, sensors data collected from a degraded aircraft engine that are generated by the National Aeronautics and Space Administration (NASA) and named C-MAPSS dataset (Saxena et al. 2008b). Finally, Section 3.6 ends the chapter with concluding remarks.

## 3.2 Proposed approach

The proposed approach is illustrated in Figure 3-1. It is splitted into two phases: offline and online. In the offline phase, a large set of features is defined and integrated into a library. Those features are issued from signal processing of measured variables by sensors. Their choice is guided by the available knowledge about the system dynamics and its components as well as the potential faults that can occur. The performances of those features in response to their application to each system will also be integrated into the library in order to facilitate their choice for new systems, in particular when there is no degradation data. Indeed, in the latter case, the choice of suitable features will be based mainly on their online RUL prediction accuracy. If some degradation data sequences are available, an RUL prediction accuracy computed in offline by using different pairs of regression models and HIs. This accuracy will

be used to select the best pairs. In the offline phase, the OCSVM is trained using the available data about normal operation conditions and the selected features. The goal is to define the best decision border that allows separating normal operation conditions zone from any potential degradation. This allows OCSVM to detect a degradation in a reliable way and in an early stage.

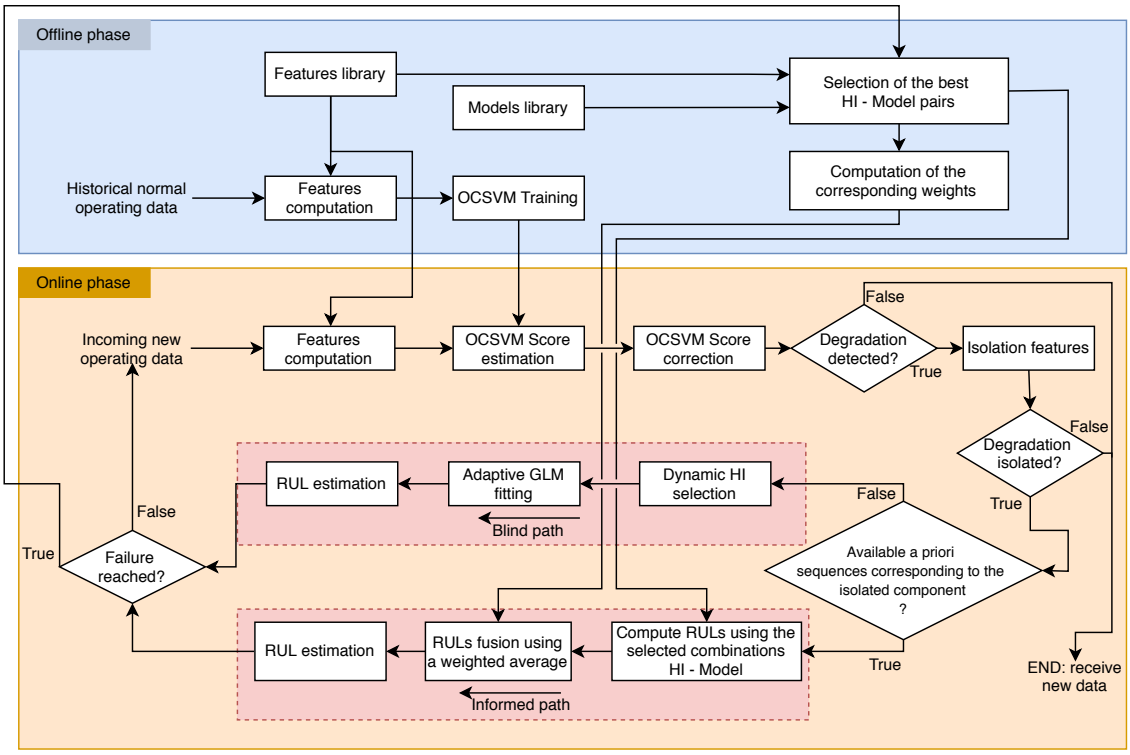



Figure 3-1: Flowchart of the proposed prognostic approach

In the online phase, OCSVM score is computed for each incoming data representing the system's current operating conditions in order to detect degradation. Fault isolation is then triggered after the detection using the isolation features. The latter are sensitive to faults in specific components. Therefore, they can be used to isolate the component responsible of the fault occurrence. Then, the RUL estimation starts by using one of two paths: the blind path or the informed path. The blind path is used when no historical degradation data are available, the RUL is estimated blindly without being confident on it. The informed path is used when some historical degradation data are available, the RUL is estimated by using some a priori knowledge about the degradation, which improves the accuracy and increases the confidence on

the predicted RUL. The main steps of the proposed approach are as follows: features definition, fault detection and isolation, RUL estimation using the blind path, and RUL estimation using the informed path. They are detailed in the next subsections.

### 3.2.1 Data processing and features definition

A library of features (health indicators) sensitive to the degradation occurrence and evolution should be predefined in the offline phase. These features are used, on the one hand, for degradation detection, and on the other hand, for characterizing its evolution over time in order to estimate the RUL. The chosen indicators in the library are dependent on the application domain.

For rotating machine degradation, vibration measurement is the most condition monitoring data used for monitoring rotating machinery, since the vibration increases due to mechanical troubles (e.g., worn bearings). Features are computed based on time domain and frequency domains in order to extract relevant health indicators from the data characterizing the degradation. Time domain features use statistical properties computed directly from the raw signal such as Root Mean Square (RMS), which describes the signal strength, peak to peak which is the distance from a negative peak to a positive peak, kurtosis which measures the probability density flatness degree of the signal, Shape factor which is computed as the ratio of RMS to the average value.

Frequency domain techniques are more effective for bearings fault diagnostic due to their ability to detect and isolate the degradation using frequency components. Fast Fourier Transform (FFT) is a common method in vibration signal analysis (Rai et al. 2007). The Frequency spectrum is computed using FFT to observe the characteristics of the vibration signal in the frequency domain. When a fault occurs, it can show the repetitive impulse period due to the contact between the rolling elements and the defective part. Other frequency domain features are computed based on the spectral kurtosis (SK) (Antoni 2006). SK is the kurtosis of the spectral component of the signal. It can deal with the transient behavior in a signal and can detect incipient fault even in the presence of noise (Saidi et al. 2017).

Table 3.1: Features library for bearing's health monitoring

| HI index | Name | Formula |
|---|---|---|
| $HI_1$ | RMS | $\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$ |
| $HI_2$ | Peak-to-peak | $max(x) - min(x)$ |
| $HI_3$ | Kurtosis | $\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^4}{\sigma^4}$ |
| $HI_4$ | Energy | $\sum_{i=1}^{n} x_i^2$ |
| $HI_5$ | Shape Factor | $\frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}}{\frac{1}{n}\sum_{i=1}^{n} \lvert x_i \rvert}$ |
| $HI_6$ | FT-mean | $\frac{1}{n}\sum_{i=1}^{n} S_i$ |
| $HI_7$ | FT-std | $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(S_i - \bar{S})}$ |
| $HI_8$ | FT-rms | $\sqrt{\frac{1}{n}\sum_{i=1}^{n} S_i^2}$ |
| $HI_9$ | FT-peak | $max(S)$ |
| $HI_{10}$ | FT-kurtosis | $\frac{\frac{1}{n}\sum_{i=1}^{n}(S_i-\bar{S})^4}{\sigma^4}$ |
| $HI_{11}$ | FT-skewness | $\frac{\frac{1}{n}\sum_{i=1}^{n}(S_i-\bar{S})^3}{\sigma^3}$ |
| $HI_{12}$ | FT-crestfactor | $\frac{max(S)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} S_i^2}}$ |
| $HI_{13}$ | SK-mean | $\frac{1}{n}\sum_{i=1}^{n} K_i$ |
| $HI_{14}$ | SK-std | $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(K_i - \bar{K})}$ |
| $HI_{15}$ | SK-rms | $\sqrt{\frac{1}{n}\sum_{i=1}^{n} K_i^2}$ |
| $HI_{16}$ | SK-peak | $max(K)$ |
| $HI_{17}$ | SK-kurtosis | $\frac{\frac{1}{n}\sum_{i=1}^{n}(K_i-\bar{K})^4}{\sigma^4}$ |

Table 3.1 sums up the features defined in the library for monitoring the bearing degradation using the vibration measurement, where $x$ is the signal in the temporal domain, $S$ is the spectral component computed using the FFT, and $K$ is the spectral kurtosis. In this library, five temporal domain features are computed on the raw signal: $HI_1$, $HI_2$, $HI_3$, $HI_4$, and $HI_5$, and seven statistical features are computed on the FFT: $HI_6$, $HI_7$, $HI_8$, $HI_9$, $HI_{10}$, $HI_{11}$, and $HI_{12}$. Also, five statistical features are computed on the spectral kurtosis: $HI_{13}$, $HI_{14}$, $HI_{15}$, $HI_{16}$, $HI_{17}$. The library of features can be modified or enriched by feedback or human experts according to the application domain.

### 3.2.2 Fault detection and isolation

#### Fault detection

In industrial systems, a considerable amount of data collected from healthy systems is available. On the other side, there is a lack of data collected about fault/degradation operation conditions or sometimes it is unavailable. For this reason, applying anomaly detection techniques is necessary in this case. One Class Support Vector Machines (OCSVM) is chosen among the different anomaly detection methods for several reasons such as it does not require any assumption about data distribution, it can deal with high dimensional data, and also it can deal with complex problems (non linear decision boundary).

OCSVM separates all the data points from the origin in a high dimensional feature space. The objective is to find an optimal hyperplane that maximizes the distance. It can be formulated as a quadratic programming problem (Schölkopf et al. 2000):

$$\min_{\omega,\xi,\rho} \quad \frac{1}{2}\|\omega\|^2 + \frac{1}{\nu.n}\sum_{i}^{n}\xi_i - \rho \tag{3.1}$$

Subject to

$$(\omega \cdot \Phi(x_i)) \geq \rho - \xi_i, \xi_i \geq 0 \tag{3.2}$$

Where $n$ is the number of training samples, $\omega$ is the normal vector separating hyperplane, $\rho$ is the offset of the desired hyperplane, $\xi = [\xi_1 ... \xi_n]$ is a vector of errors, and $\Phi(.)$ maps $x_i$ into a higher dimensional space. $\nu \in [0, 1]$ is an upper bound on the fraction of training samples outside the decision boundaries and a lower bound on the fraction of support vectors.

After solving the quadratic programming problem, the final decision function for a new point $x'$ is:

$$s(x') = \sum_{i}^{n}\alpha_i K(x_i, x') - \rho \tag{3.3}$$

$$f(x') = \text{sgn}(s(x')) \tag{3.4}$$

$$K(x_i, x') = exp[-\|x_i - x'\|^2/2\sigma^2] \tag{3.5}$$

Where $x'$ is the new sample, $x$ refers to the training points, $sgn$ is a sign function that returns +1 for positive values and -1 for the negative ones, $\alpha_i$ is the observation coefficient, $K$ is the kernel function, where Radial Basis Function (RBF) is used (Eq. (3.5)), $\sigma^2$ is the variance, and $\|.\|$ is the Euclidean norm.

The function $f(x')$ returns +1 if the observation $x'$ belongs to the known regions and -1 elsewhere. The OCSVM score $s(x')$ is computed using Eq. (3.3) which is positive in the normal class, negative outside, and 0 on the boundary. In order to improve the reliability of detection, the computed score is corrected using a moving median.

### Fault isolation

A discriminant feature (or several features) should be computed to isolate the component (or element) responsible of the fault. The isolation feature is dependent on the application domain. For example, in bearings when the rolling elements (bearing balls) pass over the defected part, they generate an impact. The successive impacts produce a series of impulse responses in the vibration signal. The spectrum of the signal can show a harmonic series of frequency components spaced at the bearing defect frequency. The latter depends on the nature of faults (e.g., inner race, outer race, ball bearings). The computed isolation feature is the median of the distance between each two successive harmonic peaks in the spectrum. After that, the computed median distance is compared with the different defects frequencies references. The latter are computed a priori using the mechanical characteristics of the bearing (e.g., ball diameter, pitch diameter, number of balls) and operating conditions (rotational speed).

### 3.2.3 Remaining useful life estimation using the blind path

The degradation detection triggers one of the RUL estimation paths depending on the availability of a priori sequences. In the case where no a priori sequences are available, the blind path is triggered because the best health indicator is unknown. Hence, the best HI is selected dynamically (selected online for each time cycle), then the best one according to a selection criterion is used for fitting a Generalized Linear Model (GLM) that allows estimating the RUL.

#### Dynamic HI selection

Since no degradation sequence is available, there is no a priori knowledge about the most sensitive HIs to the degradation evolution. For this reason, in the blind path, the best HI is selected dynamically (online) among the different features in the library according to selection criteria. The most used HI selection criteria in the literature are monotonicity and trendability (J. B. Coble 2010b; Abid et al. 2019). The monotonicity evaluates the negative or positive trend of the HI, while the trendability is related to time and represents the correlation between the HI and the operating time. However, some HIs show good trendability and monotoncity at certain time periods and then their trendability and/or monotoncity decrease significantly for other time periods. Figure 3-2 shows an example of the variable trendability and monotoncity of HIs over time. This figure illustrates the monitoring of degradation evolution using two different HIs (peak-to-peak and SK-kurtosis). The SK HI shows a very good trendability and monotonicity at the beginning of the degradation better than the peak-to-peak HI since it reacts to the degradation much time before than peak-to-peak HI. However, the SK HI's trendability and monotonicity decrease significantly while the peak-to-peak HI's trendability and monotonicity remain good. However, the SK based HI is very efficient for early degradation detection.

In order to overcome this limit, a selection criterion based on the goodness of fit is used. The HI that best fits the GLM is selected. The $R^2$ or goodness of fit indicator is used to measure how close the HI is to the fitted GLM. The higher the $R^2$ is, the

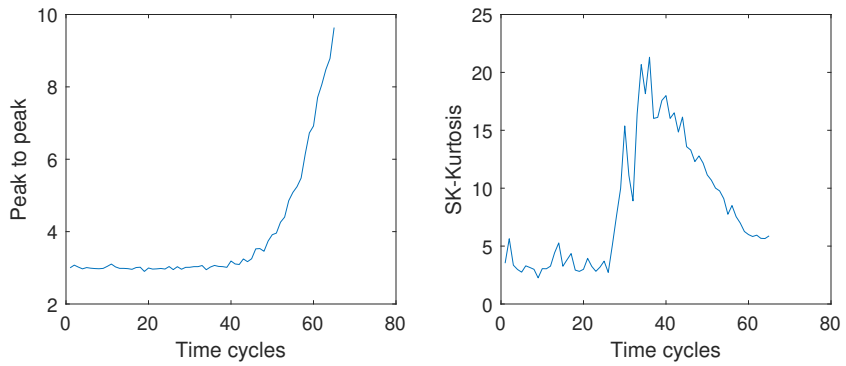



Figure 3-2: Example of two HIs evolution over time.

better the GLM fits the HI.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.6}$$

$R^2 \in [0, 1]$, where $n$ is the number of samples, $y_i$ is the HI value at index $i$, $\hat{y}_i$ is the estimated HI value using the GLM, $\bar{y}$ is the HI mean.

The GLM is fitted with the points starting from the degradation time until the present time. The HI with the highest goodness of fit $R^2$ is selected at each time cycle. The selected HI is then used to predict the degradation evolution and estimate the RUL.

### Blind RUL estimation

The RUL is the time difference between the present time and the time of failure, also named time of End Of Life $t_{EOL}$ as shown in Figure 3-3. $t_{EOL}$ is the time when the selected HI exceeds a predefined failure threshold. In this work, the Generalized Linear Model (GLM) is used to extrapolate the selected HI until it reaches the failure threshold. In the blind path, the failure threshold should be predefined by an expert.

GLM is a flexible generalization of the standard linear regression (McCullagh 2018). It can be used for different response distributions belonging to the exponential family (e.g., the normal, binomial, Poisson...). GLM generalizes linear regression by allowing the linear model $\alpha + X\beta$ to be related to the response variable via a link function ($g(\mu) = (\alpha + X\boldsymbol{\beta})$). Accordingly, the regression model is given by:

$$\mathrm{E}(\mathbf{Y}) = \mu = g^{-1}(\alpha + X\beta) \tag{3.7}$$

The coefficient estimation of the GLM is achieved using the method of maximum likelihood. The development of a GLM can be viewed as choosing the response distribution and the link function, where the selection of the appropriate parameters depends on the application.

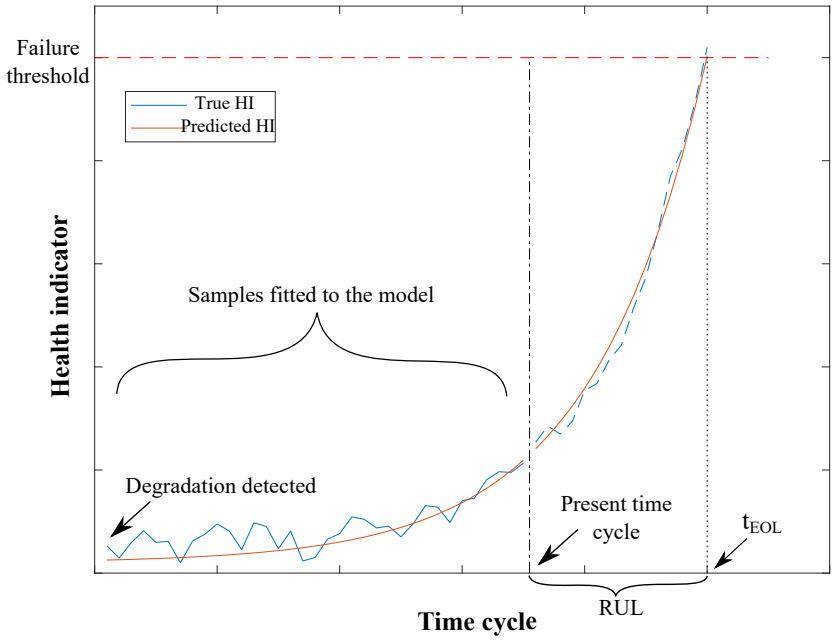



Figure 3-3: Illustration of the RUL estimation.

The GLM is used in an adaptive manner, which means that for each time cycle, the GLM is re-fitted with the present and past samples to predict the HI evolution until it reaches the failure threshold. Finding $t_{EOL}$ is necessary to estimate the RUL. The RUL estimation starts when the isolation is achieved.

### 3.2.4 Remaining useful life estimation using the informed path

This part deals with the case where few run-to-failure degradation sequences are stored offline. A degradation sequence starts from the degradation detection until the failure. The proposed method aims at taking benefit of the diversity of differ-

ent HIs and extrapolation models to estimate the RUL. The available degradation sequences are used to guide the selection of the best HI-model pairs and to determine their weights according to their performances on the RUL estimation based on those available sequences. The final RUL is computed by using a weighted mean of the estimated RULs with the selected HI-model pairs.

**HI-model pairs selection and determination of their corresponding weights**

Comparing to machine learning and deep learning techniques that require a huge amount of historical data for RUL estimation, extrapolation based models (e.g., linear regression, exponential model, quadratic regression...) are efficient even when few data are collected because they can be fitted with the collected HI samples and used to extrapolate the HI evolution until the failure (see Figure 3-4).

In this work, an ensemble of extrapolation based models is applied to extrapolate the HIs. Ensemble based method can improve the reliability and the accuracy of the prediction thanks to combining different models that are fitted with different HIs. The stored run-to-failure sequences are used to estimate the RUL using different HIs and different extrapolation based models. Then an accuracy $a_{ij}$ (Eq. (3.10)) is calculated for each RUL predicted by the HI-model pair, where $i$ represents the index of a HI, and $j$ the index of a used model:

$$a_{ij}^{s} = CRA_{ij}^{s} \tag{3.8}$$

$$CRA = \frac{1}{n_t}\sum_{t=1}^{n_t} w(t)RA(t) \tag{3.9}$$

$$RA(t) = 1 - \frac{|rul(t) - true\_rul(t)|}{true\_rul(t)} \tag{3.10}$$

$$w(t) = \frac{\frac{1}{true\_rul(t)}}{\sum_{t=1}^{n_t}\frac{1}{true\_rul(t)}} \tag{3.11}$$

Where $s$ is the sequence number, $n_t$ is the number of time index (time cycle) from

RUL estimation start until failure (EOL), and $w(t)$ is a weight factor as a function of the actual RUL (Goebel et al. 2011). The weights aim to improve the RUL evaluation by penalizing more the errors closer to $t_{EOL}$. Table 3.2 presents an example of four HIs and three models, where the accuracy $a_{ij}$ is computed for each combination HI-model. The Cumulative Relative Accuracy (CRA) is used to evaluate the RUL prediction according to the actual RUL (Saxena et al. 2008c). CRA is the weighted average of the Relative Accuracy (RA), where the RA is a measure of the error in RUL prediction ($rul$) relative to the actual RUL ($true\_rul$) at a specific time index $t$. The time index for RUL prediction $t$ starts when the degradation is isolated and stops when the HI exceeds the failure threshold at $t_{EOL}$.

Table 3.2: An example of accuracy computation for each combination HI-model

| | $\text{Model}_1$ | $\text{Model}_2$ | $\text{Model}_3$ |
|---|---|---|---|
| $\text{HI}_1$ | $a^s_{11}$ | $a^s_{12}$ | $a^s_{13}$ |
| $\text{HI}_2$ | $a^s_{21}$ | $a^s_{22}$ | $a^s_{23}$ |
| $\text{HI}_3$ | $a^s_{31}$ | $a^s_{32}$ | $a^s_{33}$ |
| $\text{HI}_4$ | $a^s_{41}$ | $a^s_{42}$ | $a^s_{43}$ |

When there are two or more run-to-failure sequences available offline, the combinations accuracies $a^s_{ij}$ (as in Table 3.2) are computed for each sequence. Then, the mean is computed between the combinations accuracies of all the run-to-failure sequences, as shown in Eq. (3.12).

$$a_{ij} = \frac{1}{n_s} \sum_{s=1}^{n_s} a^s_{ij} \tag{3.12}$$

Where $a^s_{ij}$ is the combinations accuracy for each run-to-failure sequence $s$, and $n_s$ is the number of available sequences. More the degradation sequences are available better the estimation accuracy of RUL for a new degradation sequence is. Therefore, at each time when a new degradation sequence is available, the selected HI-model pairs and their corresponding weights are updated. This is done in order to represent or cover better the variation of different degradation sequences' dynamics.

When the combinations accuracies are calculated, the HI-model combinations are ranked according to the computed accuracies $a_{ij}$. Thereafter, the best combinations are selected when the accuracy is greater than a predefined threshold. A weight $W_k$ is computed for each selected combination as a function of its accuracy. The weights can be defined as the normalization of the corresponding accuracy with a power factor $\gamma$ as follows:

$$W_k = \frac{(a_{ij})_k^{\gamma}}{\sum_{k=1}^{n_k}(a_{ij})_k^{\gamma}} \tag{3.13}$$

Where $n_k$ is the number of selected HI-model pairs, $(a_{ij})_k$ is the ranked combination accuracy where $(a_{ij})_1$ is the combination with the highest accuracy and $(a_{ij})_{n_k}$ is the lowest accuracy greater than the predefined threshold. $\gamma$ is a power factor used to give more weight for the combination with the highest accuracy. Table 3.3 is an example of 4 combinations ranked and selected (accuracy greater than a threshold). They are ranked according to the RUL estimation accuracy (the highest accuracy has the first rank), and the according weights are computed using Eq. (3.13).

Table 3.3: An example of HI-model combination selection and ranking

| Rank | Selected HI | Selected Model | Accuracies | Weights |
|---|---|---|---|---|
| 1 | $HI_2$ | $Model_1$ | $(a_{21})_1$ | $W_1$ |
| 2 | $HI_4$ | $Model_3$ | $(a_{43})_2$ | $W_2$ |
| 3 | $HI_1$ | $Model_1$ | $(a_{11})_3$ | $W_3$ |
| 4 | $HI_3$ | $Model_2$ | $(a_{32})_4$ | $W_4$ |

### Informed RUL estimation

When the HI-model combinations are selected, and their corresponding weights are computed offline, this triggers online the RUL estimation using the informed path (see Figure 3-1). In this path, the RUL is calculated using a weighted average of RULs (as shown in Figure 3-4) estimated by the selected HI-model pairs as follows:

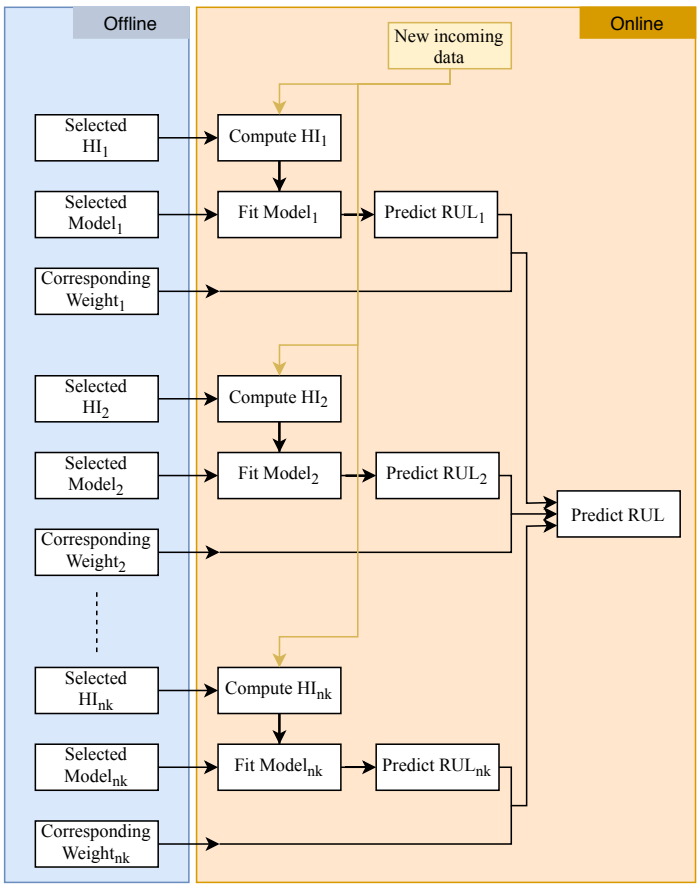



Figure 3-4: Diagram of the RUL prediction using the HI-model pairs

$$rul(t) = \sum_{k=1}^{n_k} W_k . rul(t)_k \tag{3.14}$$

Where $rul(t)$ is the final RUL estimated at each time cycle $t$, $rul(t)_k$ is the RUL estimated by the HI-model pair $k$ at each time cycle, and $W_k$ is the corresponding weight to each pair.

## 3.3 Experimentation using high speed shaft bearing degradation data

### 3.3.1 Data presentation and failure description

Vibration data are collected from a real high speed shaft bearing installed in a real commercial wind turbine with a 2MW power output provided by the Green Power Monitoring Systems in USA (Bechhoefer et al. 2013). After the last day of recording, an inspection of the bearing showed that the inner race was cracked (Figure 3-5).

Figure 3-5: Cracked inner race of the high speed shaft bearing after the last day of recording.

The run-to-failure vibration signal is measured each day for 6 seconds at a high sample rate (97656 samples per second), while this measure is repeated for 50 days. The unit of measurement is in "g", where 1g is the earth gravitational acceleration. Figure 3-6 shows the collected run to failure vibration signal over 50 days, where each part of the signal with different color has a length of 585936 samples (97656 samples times 6 seconds).

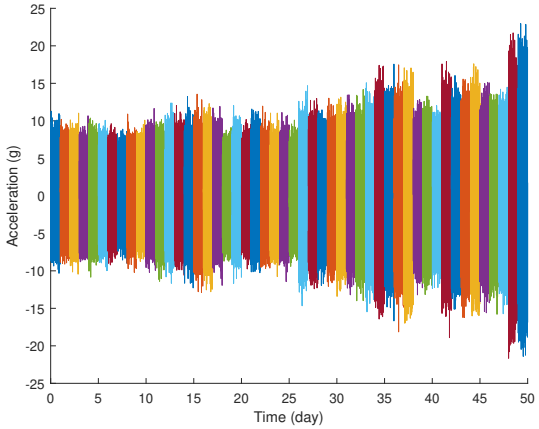



Figure 3-6: Collected vibration signal

### 3.3.2 Results and discussion

#### Data processing and features definition

The features library presented in Table 3.1 is used for this dataset since this library is suitable for vibration data. The 17 features are computed for each day (each day includes 6 seconds of measurement). The computed features are then smoothed using a moving mean with a window of 2 days in order to reduce the fluctuation and effect of noise. The computed features are illustrated in Figure 3-7. It can be observed that some features have a more monotonic trend than other features.

#### RUL estimation using the blind path

For this dataset, the RUL estimation starts directly from the beginning of the sequence, since the degradation data is a run-to-failure sequence which means that the degradation starts since the beginning of the measurement. Since there is no available a priori sequences, the blind path is triggered. First, the best HI is selected automatically and dynamically (i.e., for each day) using the selection criterion $R^2$ defined in Eq. (3.6). The selected HI for each day is shown in Figure 3-8. From this figure, it can be observed that the selected HI is varying at the beginning. However, when more data are collected, the selected HI becomes constant. The constant selected HI is the mean of the spectral kurtosis ($HI_6$ in Table 3.1), which confirms the suitability

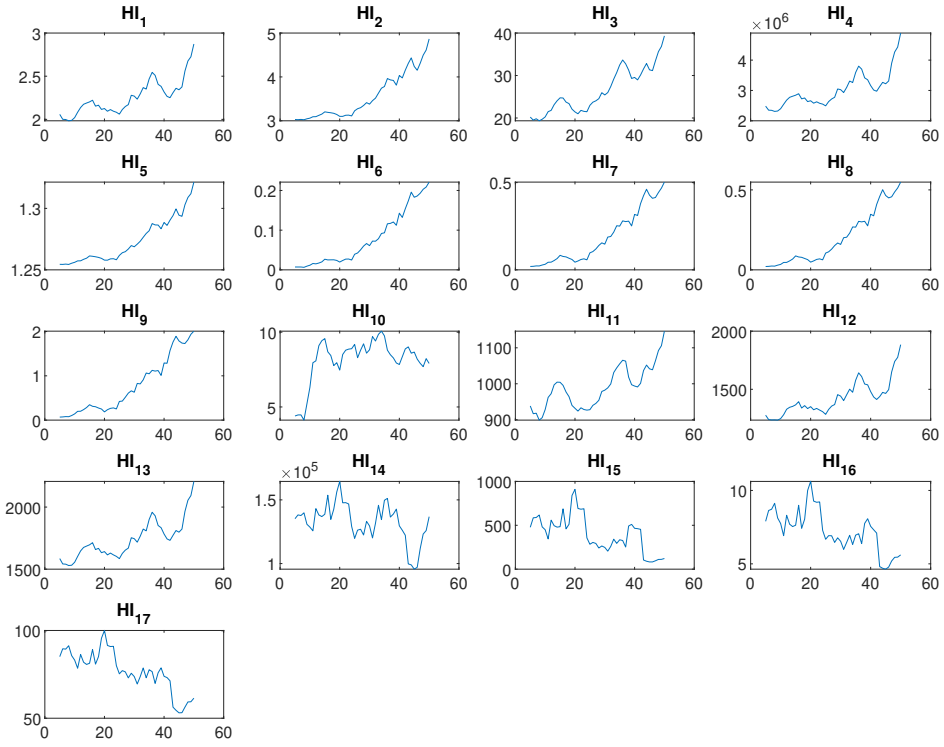



Figure 3-7: Computed indicators using HIs of Table 3.1

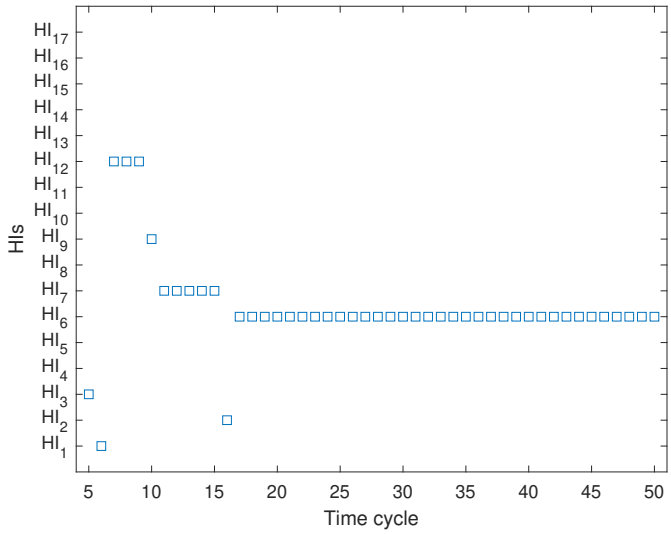
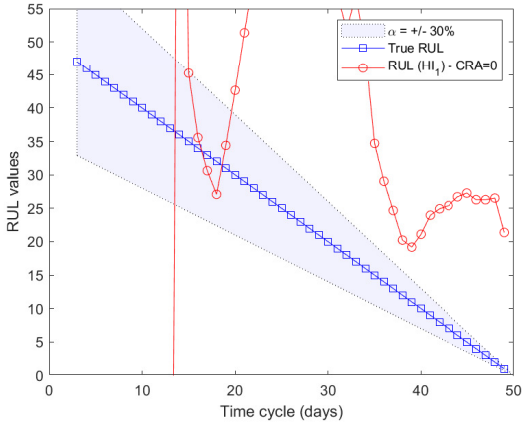


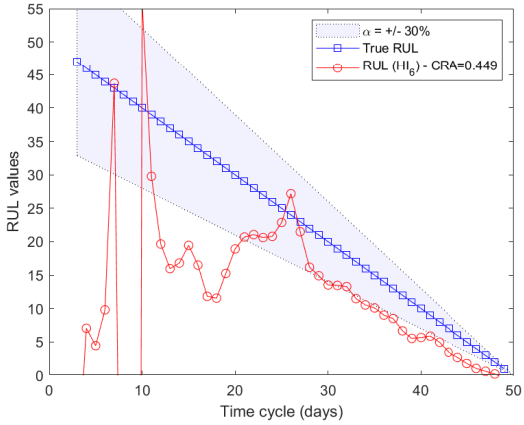

Figure 3-8: Dynamic HI selection

of spectral kurtosis based features for the health monitoring of the wind turbine shaft bearing (Saidi et al. 2017).

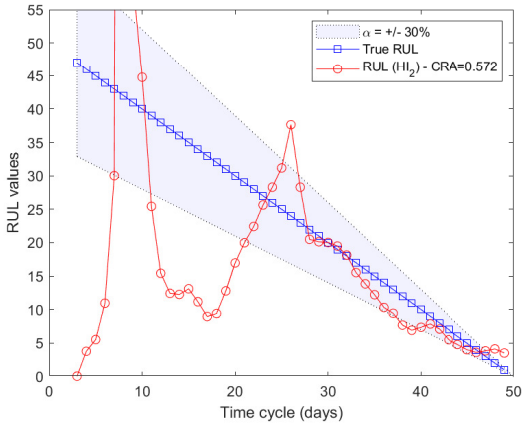
When the best HI is selected according to the selection criterion, the degradation evolution is predicted using the adaptive GLM model until the failure. Then, the RUL is computed. The predicted RUL using the proposed approach (dynamic HI selection) is shown in Figure 3-9d, it has an accuracy of $CRA = 0.504$, and it can be seen that the RUL estimation becomes more accurate with time. This is thanks to the dynamic HI selection and the adaptive GLM model that updates the model parameters with every new observation. The predicted RUL using the proposed approach is compared with the RUL estimation when using a predefined feature (without using dynamic



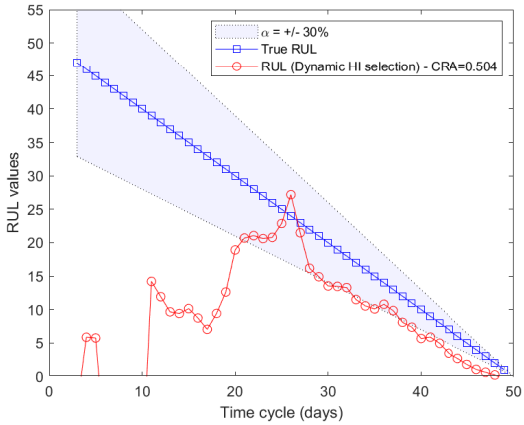

(a) Predicted RUL using predefined $HI_1$




(b) Predicted RUL using predefined $HI_6$




(c) Predicted RUL using predefined $HI_2$




(d) Predicted RUL using dynamic HI selection

Figure 3-9: RUL prediction comparison when using predefined HI and dynamic HI selection

HI selection). It can be observed that the RUL estimation accuracy by using the $HI_1$=RMS is very low (see Figure 3-9a), and the accuracy when using the $HI_6$=SK-mean ($CRA = 0.449$), which is slightly lower than using the dynamic HI selection as shown in Figure 3-9b. The time domain kurtosis $HI_2$ shows a slightly better accuracy ($CRA = 0.572$) than the dynamic HI selection (as illustrated in Figure 3-9c). However, the best HI cannot be predefined in blind RUL estimation (when no a priori sequences are available). Hence, our proposed approach using the dynamic HI selection can overcome this issue.

## 3.4 Experimentation using data of a faulty rolling bearing

### 3.4.1 Faulty bearing model presentation

Rolling bearings are commonly used in rotating machinery that permit the rotation of the shaft in the machinery. The main parts of the rolling bearing consist of (see Figure 3-10): the outer race, the inner race, and the bearing balls.

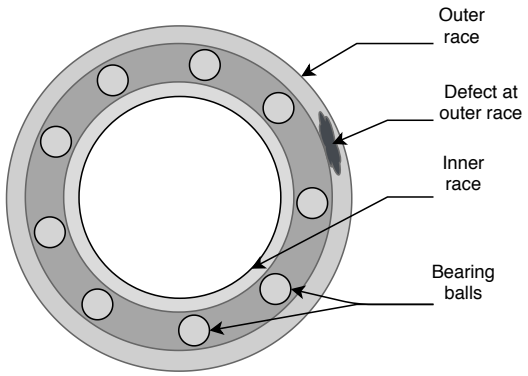



Figure 3-10: Main elements of the rolling bearing

In order to validate the proposed approach, a mathematical model representing the bearing vibratory signature within outer race defect $x_{BPFO}$ (Ball Pass Frequency outer (BPFO)), and inner race defect $x_{BPFI}$ (Ball Pass Frequency Inner (BPFI)) is used (McFadden et al. 1984; Antoni 2007). This model describes the impact produced by

the outer (inner) race fault at each passage of a rolling element at time $t$. $f_{BPFO}$ and $f_{BPFI}$ are the ball pass frequencies for the outer race and the inner race, respectively.

Table 3.4: Bearing dimensions

| | | |
|---|---|---|
| $D$ | Outer diameter | $62mm$ |
| $d$ | Inner diameter | $30mm$ |
| $D_m$ | Pitch diameter | $46mm$ |
| $n_b$ | Number of balls | 9 |
| $d_{ball}$ | Ball diameter | $9.525mm$ |
| $m$ | Modulation index | 0.7 |
| $f_0$ | Sampling frequency | $51.2kHz$ |
| $\alpha$ | Angle | $0°$ |

The simulated bearing dimensions are listed in Table 3.4. The model described in Eq. (3.15)-(3.19) is based on the bearing dimensions, and on five parameters that can influence the vibratory signature: amplitude $A$, rotational speed $f_r$, damping factor $\mu$, amplitude of the noise signal $b(t)$, and modulation shock signal $mc$. The generated signal window contains $N = 16348$ samples with a sampling frequency of $f_0 = 51.2kHz$.

$$x_{BPFO}(t) = \sum_{k=1}^{N} A.exp\left(-2\pi\mu f_0\left(t - \frac{k}{f_{BPFO}}\right)\right) .sin\left(2\pi f_0\left(t - \frac{k}{f_{BPFO}}\right)\right) + b(t) \tag{3.15}$$

$$f_{BPFO} = \frac{nb}{2} fr\left(1 + \frac{d_{ball}}{D_m}.cos(\alpha)\right) \tag{3.16}$$

$$x_{BPFI}(t) = \sum_{k=1}^{N} A.mc.exp\left(-2\pi\mu f_0\left(t - \frac{k}{f_{BPFI}}\right)\right) .sin\left(2\pi f_0\left(t - \frac{k}{f_{BPFI}}\right)\right) + b(t) \tag{3.17}$$

$$f_{BPFI} = \frac{nb}{2} fr\left(1 - \frac{d_{ball}}{D_m}.cos(\alpha)\right) \tag{3.18}$$

$$mc = 1 + m.cos(2\pi f_r t) \tag{3.19}$$

The generated signatures for the inner/outer race fault are shown in Figure 3-11. Outer race fault is characterized by a constant amplitude modulation, while inner race fault is characterized by a periodic amplitude modulation.

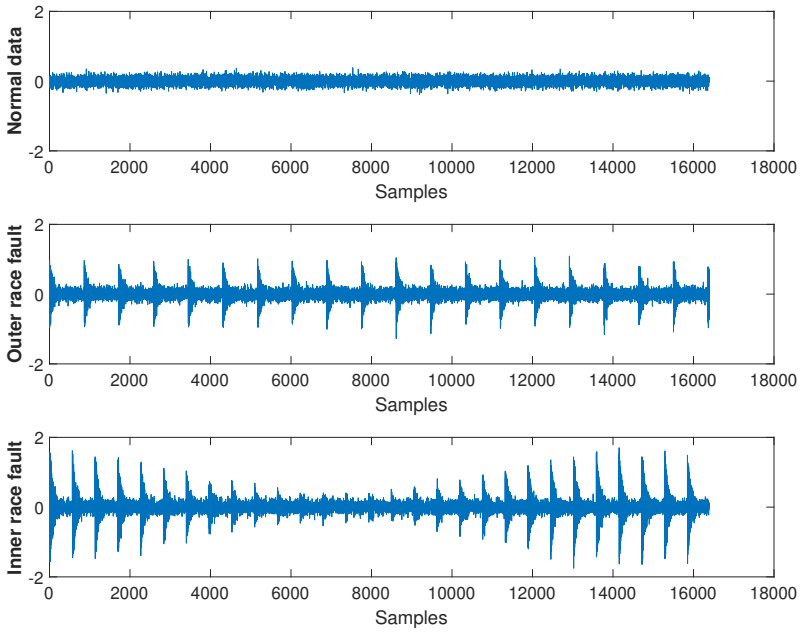



Figure 3-11: Generated fault signatures for the inner and outer race.

### 3.4.2 Fault sequences generation

By using the faulty bearing model described in the previous subsection, the amplitude $A$ of the vibration signature is increased in an exponential manner, since the degradation evolution of bearings shows an exponential growth (Gebraeel et al. 2004). Thus, for each signal window (showed in Figure 3-11), the amplitude $A$ is modified. The incipient degradation starts at the first signal window, where the failure is reached at the last signal window. In the next, the term "time cycle" is used instead of "signal window". The generated bearing degradation data are available online (Abid et al. 2020).

For each fault mode (inner or outer race fault), twenty sequences are generated to validate our proposed approach (see Table 3.5). Five degradation speeds are simulated

Table 3.5: Generated degradation sequences

| Outer/Inner race fault sequence # | Degradation speeds | Time cycle length |
|---|---|---|
| 1<br>2<br>3<br>4 | Very fast | 30 |
| 5<br>6<br>7<br>8 | Fast | 50 |
| 9<br>10<br>11<br>12 | Medium | 65 |
| 13<br>14<br>15<br>16 | Slow | 80 |
| 17<br>18<br>19<br>20 | Very slow | 100 |

(very fast, fast, medium, slow, and very slow). For each degradation speed, four sequences are generated with a variability of ±5% on the condition parameters (the five parameters cited in the previous subsection). The latter are varied to simulate the variability of the operating conditions that affect the component in reality. In order to train the OCSVM model, a sequence of normal operating data is also generated.

### 3.4.3 Results and discussion

#### Degradation detection and isolation

The features predefined in the library are computed for each time cycle (signal window). The OCSVM model is trained offline using the normal operating data in order to construct a boundary on the normal data. For a new collected signal, features are computed for each time cycle and then fed to the OCSVM model to predict the OCSVM score. The degradation is detected when the corrected OCSVM score is below the boundary (score=0). Figure 3-13 shows the degradation detection for the sequence #9 (sequence with a medium degradation speed). This sequence is used to illustrate the application of the proposed approach to the generated bearings scenar-

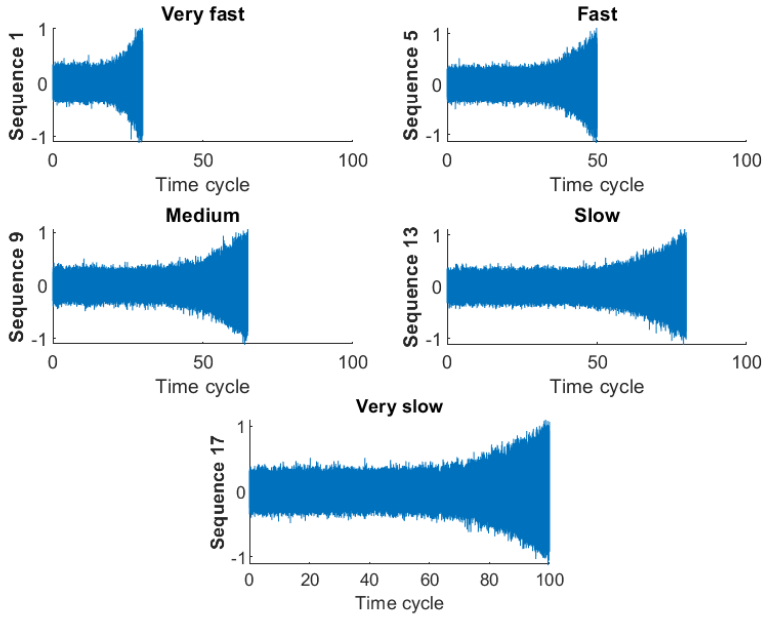



Figure 3-12: Examples of generated sequences with several degradation speeds.

ios.

Fault isolation starts when the degradation is detected, and the isolation feature (the median of the distance between each two successive harmonic peaks in the spectrum) is computed for each new window signal. Figure 3-14 shows the harmonics of the signal in the frequency domain. The peaks of the harmonics greater than a peak threshold ($P_{th}$) are selected. $P_{th} = 48$ is selected using $\mu + 3.\sigma$ rule on the peaks of the normal operating signal ($\mu$ is the mean and $\sigma$ is the STD of the normal signal peaks), then the frequency distances between the peaks are computed. In order to isolate the fault, the median of the computed frequency distances is compared with the bearing defects frequencies $f_{BPFO} = 59.47Hz$ and $f_{BPFI} = 90.52Hz$, which are computed offline using the mechanical characteristics of the bearing by applying Eq. (3.16) and Eq. (3.18). For each new window the $dist_{med}$ is computed, the fault is then isolated when it is within the defect frequency boundary $\pm 10\%$ ($f_{defect} \pm 0.1.f_{defect}$).

Degradation detection and fault isolation time results are presented in Table 3.6. It can be seen that the detection time is correlated with the degradation speed, because when the degradation is slow, it is difficult to detect the fault early. Then,

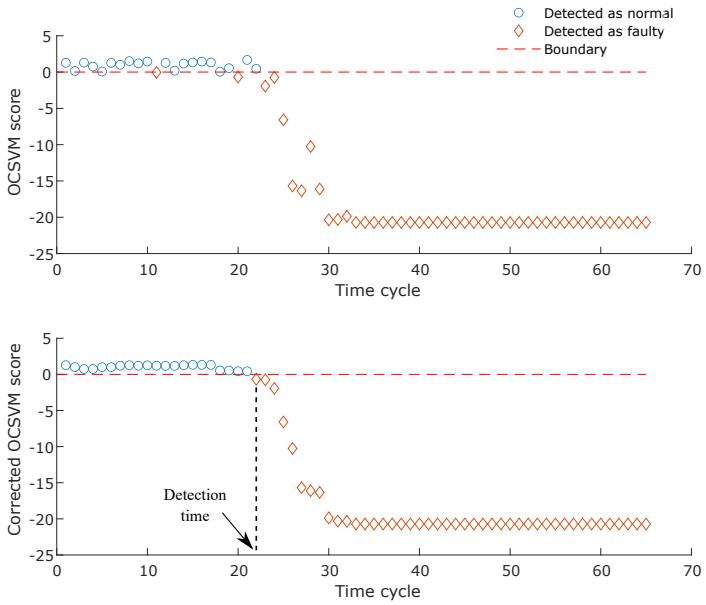



Figure 3-13: Degradation detection for sequence 9.

when the detection time percent is computed (i.e., the ratio of detection time to the sequence length), it can be seen that the detection percent is almost the same for all the sequence (around 35%), which means that the degradation is detected in the 35% of the sequence length. A delay is observed between detection time and isolation time due to the difficulty to observe the peaks in the frequency domain when the degradation amplitude is low, which is due to the effect of noise on the signal.

### RUL estimation using the blind path

When no a priori sequences are available, the blind path is triggered. The degradation isolation triggers the dynamic HI selection and RUL estimation steps. The collected data starting from the degradation detection can be used to start fitting the model for the RUL estimation. As mentioned in subsection 3.2.3, in this path, the HI is dynamically selected among the different features available in the library by using the selection criterion defined in Eq. (3.6) (goodness of fit by the GLM). Figure 3-15 shows the different HIs selected for the sequence #9 with outer race fault at each time index from the isolation time cycle until the failure. Then, the selected HI at each time cycle is used for RUL estimation.

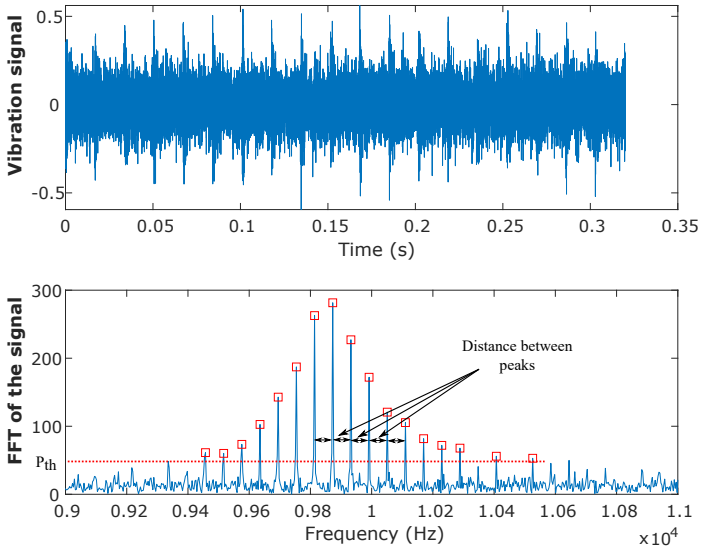



Figure 3-14: Example of computation of the distance between peaks

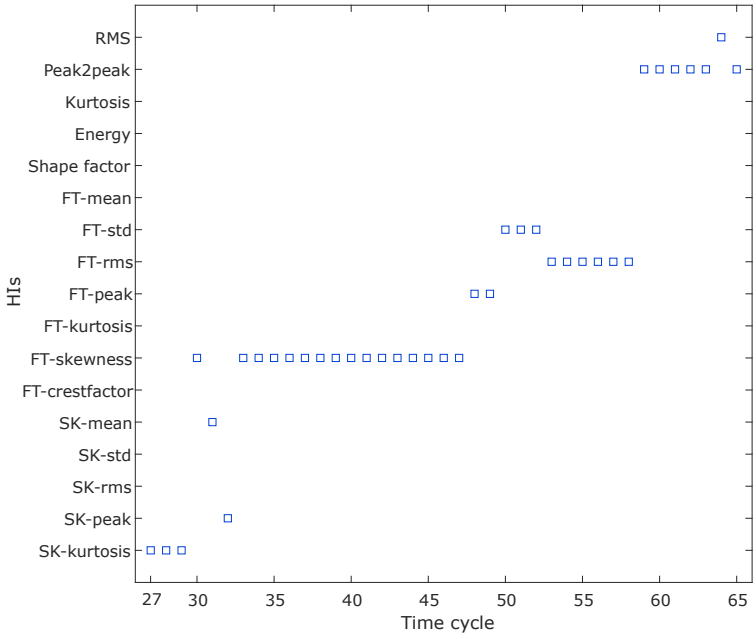



Figure 3-15: Online selected HIs for sequence #9.

The estimated RUL for the sequence #9 is presented in Figure 3-16. It shows the predicted RUL, actual RUL, and $\alpha = 0.3$ boundary, which allows the deviation of 30% from the true RUL at each time. It can be seen that the RUL becomes more accurate over time, this is due to our adaptive model that is updated with each new collected sample. The RUL estimation accuracies using all the generated fault

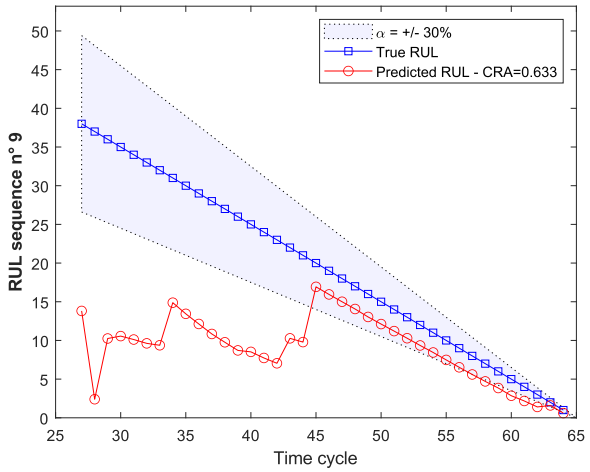



Figure 3-16: RUL estimation for sequence #9 with outer race fault using the blind path.

Table 3.6: Blind path RUL accuracies of the sequences with outer race fault

| Sequence # | Detection time | isolation time | Detection percent (%) | Fault type | Accuracy (Blind) |
|---|---|---|---|---|---|
| 1 | 10 | 15 | 33.3 | BPFO | 0.694 |
| 2 | 11 | 15 | 36.6 | BPFO | 0.703 |
| 3 | 11 | 16 | 36.6 | BPFO | 0.661 |
| 4 | 11 | 13 | 36.6 | BPFO | 0.698 |
| 5 | 19 | 23 | 38.0 | BPFO | 0.655 |
| 6 | 17 | 23 | 34.0 | BPFO | 0.661 |
| 7 | 17 | 23 | 34.0 | BPFO | 0.670 |
| 8 | 18 | 22 | 36.0 | BPFO | 0.692 |
| 9 | 22 | 27 | 33.8 | BPFO | 0.633 |
| 10 | 24 | 29 | 36.9 | BPFO | 0.687 |
| 11 | 23 | 27 | 35.3 | BPFO | 0.591 |
| 12 | 21 | 30 | 32.3 | BPFO | 0.745 |
| 13 | 27 | 36 | 33.7 | BPFO | 0.614 |
| 14 | 24 | 34 | 30.0 | BPFO | 0.680 |
| 15 | 28 | 37 | 35.0 | BPFO | 0.609 |
| 16 | 28 | 37 | 35.0 | BPFO | 0.741 |
| 17 | 38 | 45 | 38.0 | BPFO | 0.616 |
| 18 | 36 | 42 | 36.0 | BPFO | 0.606 |
| 19 | 37 | 46 | 37.0 | BPFO | 0.660 |
| 20 | 36 | 42 | 36.0 | BPFO | 0.633 |

sequences for the outer race fault with the blind path are presented in Table 3.6. The overall accuracy for the blind path is 0.662, which is computed with the mean of the last column of Table 3.6. This RUL estimation accuracy is suitable because the RUL is estimated in a blind manner without a priori degradation data. This accuracy is compared in the next subsection when there are few historical degradation data.

### RUL estimation using the informed path

When few run-to-failure sequences are stored offline, the HI-model pairs are selected offline by computing the RUL accuracy using different combinations HI-model. The tested HIs are the features defined in the feature library. The used models are the widely used extrapolation based models for bearing degradation modelling: $\text{Model}_1$ : GLM (Abid et al. 2019), $\text{Model}_2$ : quadratic regression (Ahmad et al. 2017), $\text{Model}_3$ : exponential model (Gebraeel et al. 2004), and $\text{Model}_4$ : double exponential model (Jin et al. 2016; L. Guo et al. 2017). The accuracy defined in Eq. (3.8), (3.9), (3.10), and (3.11) is calculated for each HI-model pairs, then the pairs with an accuracy greater than a threshold are selected. The threshold selected is the overall accuracy of the blind path in order to choose HI-model pairs that yield RUL accuracy greater than the blind path accuracy (0.662). Thus, all the HI-model pairs greater than 0.662 are selected.

The corresponding weight for each HI-model combination is calculated using Eq. (3.13). An example of the computed accuracies offline using different models and different HIs is presented in Table 3.7, where the bold numbers represent the accuracies greater than the threshold (0.622). When the error of prediction is very high, the CRA accuracy (see Eq. (3.9)) value becomes negative. Hence, the accuracies with negative values are substituted by zero. From Table 3.7, it can be seen that

Table 3.7: Example of HI-model pairs selection

| | $\text{Model}_1$ GLM | $\text{Model}_2$ Quad Reg | $\text{Model}_3$ Exp | $\text{Model}_4$ Double Exp |
|---|---|---|---|---|
| $\text{HI}_1$ | 0.067 | 0.000 | 0.358 | 0.000 |
| $\text{HI}_2$ | 0.145 | 0.000 | 0.181 | 0.000 |
| $\text{HI}_3$ | 0.552 | 0.000 | 0.566 | 0.000 |
| $\text{HI}_4$ | 0.000 | 0.000 | 0.194 | 0.000 |
| $\text{HI}_5$ | 0.000 | 0.000 | 0.000 | 0.000 |
| $\text{HI}_6$ | 0.000 | 0.000 | 0.000 | 0.000 |
| $\text{HI}_7$ | **0.752** | 0.059 | **0.773** | 0.000 |
| $\text{HI}_8$ | 0.064 | 0.000 | 0.355 | 0.000 |
| $\text{HI}_9$ | 0.476 | 0.461 | 0.650 | **0.714** |
| $\text{HI}_{10}$ | 0.287 | 0.616 | 0.411 | 0.178 |
| $\text{HI}_{11}$ | 0.256 | 0.628 | 0.274 | 0.214 |
| $\text{HI}_{12}$ | 0.256 | **0.754** | 0.332 | 0.282 |
| $\text{HI}_{13}$ | 0.616 | 0.364 | **0.768** | 0.424 |
| $\text{HI}_{14}$ | 0.226 | **0.814** | 0.362 | 0.165 |
| $\text{HI}_{15}$ | 0.259 | **0.870** | 0.398 | 0.216 |
| $\text{HI}_{16}$ | 0.291 | 0.013 | 0.576 | 0.000 |
| $\text{HI}_{17}$ | 0.000 | 0.000 | 0.000 | 0.000 |

the HI-model pairs with a high accuracy are $HI_{15}$-$model_2$, $HI_{14}$-$model_2$, $HI_7$-$model_3$, $HI_{13}$-$model_3$, $HI_{12}$-$model_2$, $HI_7$-$model_1$, and $HI_9$-$model_4$. These HI-model pairs are selected, and their corresponding weights are calculated as a function of the accuracy, the parameter $\gamma$ has been chosen empirically ($\gamma = 8$), where more important weights are given to the pairs with high accuracy as shown in Table 3.8.

Table 3.8: Example of selected HI-model and corresponding weights

| Rank | HI | Model | Accuracy | Weight |
|---|---|---|---|---|
| 1 | 15 | 2 | 0.870 | 0.314 |
| 2 | 14 | 2 | 0.814 | 0.185 |
| 3 | 7 | 3 | 0.773 | 0.122 |
| 4 | 13 | 3 | 0.768 | 0.116 |
| 5 | 12 | 2 | 0.754 | 0.100 |
| 6 | 7 | 1 | 0.752 | 0.098 |
| 7 | 9 | 4 | 0.714 | 0.064 |

When the informed path is triggered in the online phase, different RULs are predicted using the selected HI-model pairs. After that, the computed RULs are merged using a weighted sum as defined in Eq. (3.14) to predict the final RUL. It is worth mentioning that the extrapolation model parameters are updated with each new time cycle, which allows the adaptability of our proposed approach to the different degradation speeds. In the aim to validate the proposed approach for the informed path, the RUL is predicted for all the sequences in the scenario (20 sequences) using as a priori different number and combinations of the available sequences, where the number of available a priori sequence is from zero (blind) to six sequences. In order to show the robustness of the proposed method, 20 combinations of a priori sequences are selected randomly for two until six sequences. The goal is twofold: observing the improvement of the RUL accuracy estimation with the number of available degradation sequences, and highlighting the interest of the use of the best HI-model pairs when the degradation dynamics (speed) of a new sequence is significantly different of the one in the available degradation sequences.

Each accuracy table is computed for a specific number of a priori degradation sequences (i.e., from 1 to 6). As an example Table 3.9 shows CRA for each incoming sequence (from 1 to 20) for different combinations of 3 a priori sequence (selected

Table 3.9: RUL estimation accuracy for the outer race fault with 3 a priori sequences

| | Randomly selected a priori sequences (3 sequences) | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Incoming sequences | 5<br>9<br>18 | 7<br>10<br>19 | 3<br>4<br>12 | 5<br>6<br>9 | 10<br>18<br>19 | 11<br>12<br>14 | 1<br>13<br>14 | 2<br>7<br>11 | 1<br>5<br>13 | 4<br>8<br>9 | 1<br>3<br>20 | 4<br>9<br>16 | 6<br>12<br>13 | 2<br>16<br>20 | 6<br>17<br>18 | 5<br>15<br>19 | 1<br>3<br>13 | 5<br>10<br>18 | 2<br>12<br>20 | 4<br>5<br>19 |
| 1 | 0.825 | 0.847 | 0.786 | 0.856 | 0.796 | 0.869 | 0.888 | 0.907 | 0.879 | 0.890 | 0.886 | 0.824 | 0.758 | 0.836 | 0.759 | 0.854 | 0.888 | 0.837 | 0.836 | 0.776 |
| 2 | 0.640 | 0.779 | 0.894 | 0.820 | 0.543 | 0.839 | 0.796 | 0.927 | 0.742 | 0.913 | 0.811 | 0.884 | 0.744 | 0.938 | 0.630 | 0.793 | 0.806 | 0.632 | 0.928 | 0.802 |
| 3 | 0.698 | 0.744 | 0.838 | 0.746 | 0.686 | 0.791 | 0.752 | 0.781 | 0.729 | 0.727 | 0.821 | 0.799 | 0.712 | 0.796 | 0.692 | 0.762 | 0.744 | 0.710 | 0.802 | 0.791 |
| 4 | 0.759 | 0.806 | 0.918 | 0.816 | 0.709 | 0.868 | 0.891 | 0.892 | 0.839 | 0.901 | 0.840 | 0.859 | 0.724 | 0.842 | 0.692 | 0.837 | 0.861 | 0.769 | 0.848 | 0.883 |
| 5 | 0.844 | 0.856 | 0.905 | 0.855 | 0.829 | 0.893 | 0.865 | 0.865 | 0.857 | 0.804 | 0.841 | 0.815 | 0.788 | 0.840 | 0.792 | 0.881 | 0.861 | 0.861 | 0.840 | 0.875 |
| 6 | 0.806 | 0.867 | 0.859 | 0.880 | 0.803 | 0.889 | 0.860 | 0.875 | 0.838 | 0.836 | 0.923 | 0.884 | 0.872 | 0.887 | 0.836 | 0.874 | 0.862 | 0.815 | 0.891 | 0.842 |
| 7 | 0.908 | 0.931 | 0.937 | 0.934 | 0.911 | 0.938 | 0.899 | 0.930 | 0.874 | 0.892 | 0.942 | 0.888 | 0.836 | 0.920 | 0.906 | 0.916 | 0.890 | 0.903 | 0.901 | 0.876 |
| 8 | 0.834 | 0.845 | 0.793 | 0.824 | 0.804 | 0.776 | 0.808 | 0.799 | 0.846 | 0.795 | 0.832 | 0.765 | 0.819 | 0.777 | 0.797 | 0.834 | 0.837 | 0.838 | 0.756 | 0.839 |
| 9 | 0.867 | 0.858 | 0.815 | 0.888 | 0.829 | 0.832 | 0.828 | 0.838 | 0.859 | 0.801 | 0.865 | 0.847 | 0.908 | 0.868 | 0.826 | 0.868 | 0.861 | 0.873 | 0.844 | 0.814 |
| 10 | 0.811 | 0.851 | 0.926 | 0.845 | 0.817 | 0.862 | 0.839 | 0.839 | 0.821 | 0.847 | 0.934 | 0.870 | 0.834 | 0.867 | 0.826 | 0.859 | 0.831 | 0.821 | 0.869 | 0.889 |
| 11 | 0.833 | 0.850 | 0.891 | 0.843 | 0.809 | 0.869 | 0.864 | 0.865 | 0.833 | 0.871 | 0.778 | 0.871 | 0.792 | 0.859 | 0.784 | 0.859 | 0.838 | 0.844 | 0.871 | 0.878 |
| 12 | 0.821 | 0.863 | 0.886 | 0.891 | 0.809 | 0.914 | 0.857 | 0.885 | 0.808 | 0.856 | 0.957 | 0.922 | 0.896 | 0.929 | 0.826 | 0.879 | 0.836 | 0.830 | 0.928 | 0.864 |
| 13 | 0.907 | 0.848 | 0.655 | 0.843 | 0.890 | 0.720 | 0.769 | 0.739 | 0.875 | 0.741 | 0.639 | 0.747 | 0.901 | 0.761 | 0.869 | 0.818 | 0.846 | 0.905 | 0.726 | 0.746 |
| 14 | 0.816 | 0.856 | 0.914 | 0.860 | 0.801 | 0.926 | 0.836 | 0.858 | 0.789 | 0.826 | 0.923 | 0.857 | 0.855 | 0.870 | 0.798 | 0.868 | 0.809 | 0.827 | 0.895 | 0.872 |
| 15 | 0.763 | 0.808 | 0.872 | 0.820 | 0.733 | 0.883 | 0.847 | 0.869 | 0.792 | 0.836 | 0.903 | 0.855 | 0.771 | 0.848 | 0.723 | 0.835 | 0.813 | 0.778 | 0.870 | 0.809 |
| 16 | 0.870 | 0.886 | 0.815 | 0.934 | 0.839 | 0.915 | 0.907 | 0.919 | 0.875 | 0.862 | 0.897 | 0.938 | 0.908 | 0.945 | 0.846 | 0.902 | 0.903 | 0.880 | 0.937 | 0.815 |
| 17 | 0.847 | 0.858 | 0.772 | 0.831 | 0.843 | 0.759 | 0.765 | 0.777 | 0.821 | 0.788 | 0.786 | 0.817 | 0.875 | 0.821 | 0.850 | 0.828 | 0.812 | 0.844 | 0.800 | 0.813 |
| 18 | 0.806 | 0.754 | 0.612 | 0.765 | 0.831 | 0.704 | 0.707 | 0.699 | 0.757 | 0.672 | 0.702 | 0.686 | 0.826 | 0.703 | 0.843 | 0.734 | 0.740 | 0.799 | 0.685 | 0.653 |
| 19 | 0.884 | 0.897 | 0.842 | 0.879 | 0.897 | 0.834 | 0.833 | 0.850 | 0.869 | 0.843 | 0.761 | 0.849 | 0.890 | 0.864 | 0.891 | 0.878 | 0.863 | 0.889 | 0.851 | 0.862 |
| 20 | 0.810 | 0.866 | 0.918 | 0.879 | 0.794 | 0.923 | 0.877 | 0.920 | 0.833 | 0.863 | 0.952 | 0.903 | 0.849 | 0.917 | 0.801 | 0.886 | 0.854 | 0.824 | 0.915 | 0.877 |

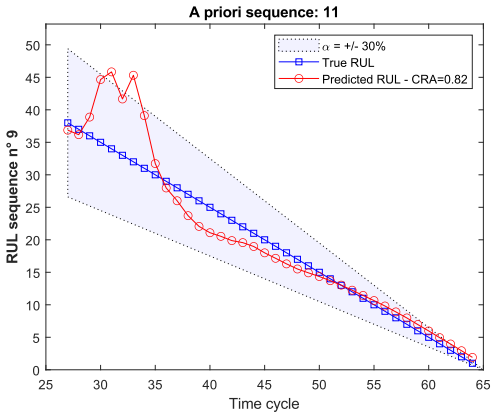
Table 3.10: RUL estimation accuracy for the outer race fault using the informed path with different number of a priori sequences

| | Number of a priori sequences | | | | | |
|---|---|---|---|---|---|---|
| **Incoming sequences** | **CRA (1 seq)** | **CRA (2 seqs)** | **CRA (3 seqs)** | **CRA (4 seqs)** | **CRA (5 seqs)** | **CRA (6 seqs)** |
| 1 | 0.718±0.173 | 0.811±0.099 | 0.840±0.045 | 0.855±0.068 | 0.853±0.050 | 0.860±0.035 |
| 2 | 0.728±0.134 | 0.807±0.077 | 0.793±0.112 | 0.844±0.078 | 0.846±0.082 | 0.826±0.080 |
| 3 | 0.783±0.087 | 0.782±0.044 | 0.756±0.045 | 0.770±0.037 | 0.765±0.033 | 0.754±0.025 |
| 4 | 0.667±0.249 | 0.782±0.179 | 0.828±0.066 | 0.853±0.088 | 0.862±0.055 | 0.848±0.053 |
| 5 | 0.700±0.367 | 0.800±0.167 | 0.848±0.031 | 0.838±0.123 | 0.860±0.031 | 0.858±0.018 |
| 6 | 0.820±0.080 | 0.862±0.037 | 0.860±0.031 | 0.863±0.024 | 0.861±0.022 | 0.865±0.015 |
| 7 | 0.870±0.065 | 0.905±0.027 | 0.906±0.026 | 0.921±0.017 | 0.913±0.023 | 0.920±0.016 |
| 8 | 0.743±0.140 | 0.804±0.041 | 0.811±0.028 | 0.804±0.027 | 0.804±0.025 | 0.808±0.022 |
| 9 | 0.831±0.097 | 0.856±0.049 | 0.849±0.027 | 0.840±0.036 | 0.843±0.036 | 0.852±0.032 |
| 10 | 0.762±0.172 | 0.820±0.118 | 0.853±0.033 | 0.848±0.026 | 0.851±0.026 | 0.847±0.018 |
| 11 | 0.692±0.282 | 0.790±0.150 | 0.845±0.032 | 0.833±0.105 | 0.858±0.019 | 0.856±0.017 |
| 12 | 0.797±0.149 | 0.853±0.088 | 0.873±0.043 | 0.871±0.042 | 0.881±0.027 | 0.886±0.017 |
| 13 | 0.681±0.215 | 0.744±0.122 | 0.797±0.083 | 0.754±0.105 | 0.774±0.074 | 0.796±0.065 |
| 14 | 0.720±0.253 | 0.809±0.135 | 0.853±0.041 | 0.844±0.081 | 0.870±0.039 | 0.865±0.025 |
| 15 | 0.718±0.192 | 0.804±0.079 | 0.821±0.050 | 0.832±0.057 | 0.839±0.035 | 0.835±0.031 |
| 16 | 0.804±0.169 | 0.872±0.086 | 0.890±0.039 | 0.885±0.071 | 0.898±0.036 | 0.912±0.026 |
| 17 | 0.780±0.093 | 0.801±0.051 | 0.815±0.033 | 0.788±0.044 | 0.791±0.045 | 0.801±0.041 |
| 18 | 0.688±0.125 | 0.715±0.045 | 0.734±0.063 | 0.708±0.045 | 0.716±0.056 | 0.732±0.047 |
| 19 | 0.749±0.248 | 0.816±0.123 | 0.861±0.031 | 0.832±0.098 | 0.852±0.021 | 0.859±0.023 |
| 20 | 0.820±0.090 | 0.882±0.041 | 0.873±0.045 | 0.886±0.032 | 0.886±0.032 | 0.882±0.025 |
| **Overall accuracy** | **0.753±0.169** | **0.816±0.088** | **0.835±0.045** | **0.833±0.060** | **0.841±0.038** | **0.843±0.032** |

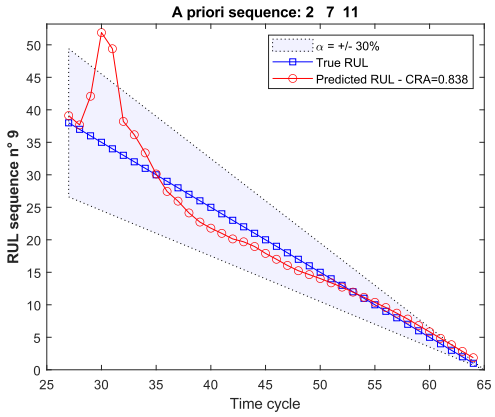
randomly). Then, the average and the standard deviation of each row of the accuracy tables (such as in Table 3.9 for the case of 3 a priori sequence) is computed and shown in Table 3.10.

The average is computed for those accuracy tables of accuracies with different a priori sequences, from one a priori sequence until six a priori sequences. The columns of Table 3.10 present the average and the standard deviations of the accuracy tables.

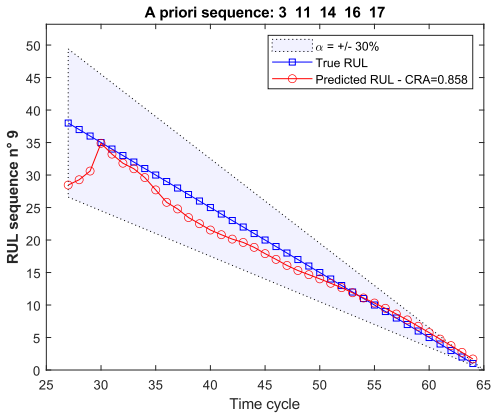
For example, the average and the standard deviation of each row in Table 3.9 is shown in the column (named CRA 3 seqs) of Table 3.10. The overall accuracy is then computed as the average of each column of Table 3.10, it is computed with the objective to show the overall accuracy improvement when more a priori sequences are available. Table 3.10 shows that the overall accuracy is improved when more a priori sequences are available. The computed accuracies when there are more than six a priori sequences are not showed because for more than six a priori sequences, no significant improvement in the RUL estimation accuracies is observed.




(a) Predicted RUL using available a priori sequence #11




(b) Predicted RUL using available a priori sequences #2, #7, and #11




(c) Predicted RUL using available a priori sequences #3, #11, #14, #16, and #17

Figure 3-17: RUL prediction for the sequence #9 using different available a priori sequences

In the aim to validate the robustness of the proposed approach, Table 3.10 presents

as well the standard deviation (STD) of the accuracies. It can be seen that the overall STD is decreasing when more a priori sequences are available, this confirms that the proposed approach is more robust while having more a priori sequences.

Figure 3-17 shows the predicted and the true (actual) RUL for the sequence #9 using different number of a priori sequences. One sequence (#11) is used as a priori in Figure 3-17a, three a priori sequences (#2, #7, and #11) are used in Figure 3-17b, and five sequences (#3, #11, #14, #16, and #17) are used in Figure 3-17c. From these figures, it can be observed that the RUL prediction accuracy is improved when more a priori sequences are available.

Ensemble modelling allows the diversity exploitation of the different models that can enhance the accuracy of prediction, which means that using several models is better than the use of one model. In order to confirm this, the RUL is computed using only the top best model in the selected HI-model pairs. For example, the RUL is predicted using only the top best HI-model pair in Table 3.8. Thus, the RUL is predicted using $\text{model}_2$ = Quadratic Regression fitted by $\text{HI}_{15}$ = SK-rms. The validation is the same as for the proposed approach using an ensemble of models, where the results are presented in Table 3.11. This table shows that the accuracy using an ensemble of models is better than using only the top best model (one model). Table 3.11 presents a comparison according to the overall RUL estimation accuracy of a faulty bearing with an outer race fault.

Table 3.11: RUL overall accuracy comparison for the outer race fault

| | CRA Blind | CRA 1seq | CRA 2seq | CRA 3seq | CRA 4seq | CRA 5seq | CRA 6seq |
|---|---|---|---|---|---|---|---|
| SVR | - | 0.366 | 0.529 | 0.575 | 0.634 | 0.655 | 0.675 |
| LSTM | - | 0.316 | 0.586 | 0.698 | 0.767 | 0.831 | **0.852** |
| Best HI-model | **0.662** | 0.664 | 0.791 | 0.789 | 0.799 | 0.819 | 0.822 |
| Proposed approach | **0.662** | **0.753** | **0.816** | **0.835** | **0.833** | **0.841** | 0.843 |

The proposed approach using an ensemble of HI-model pairs is compared with using only the first best HI-model pair, and with two other methods: SVR and LSTM. SVR has proven its effectiveness for RUL estimation of bearings (Benkedjouh et al. 2013; Saidi et al. 2017; Soualhi et al. 2014). It is a machine learning regression method

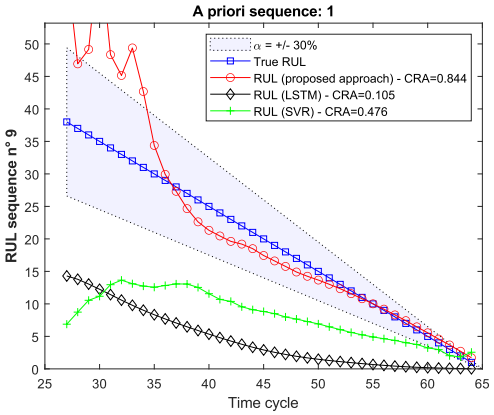
Table 3.12: RUL overall accuracy comparison for the inner race fault

| | CRA Blind | CRA 1seq | CRA 2seq | CRA 3seq | CRA 4seq | CRA 5seq | CRA 6seq |
|---|---|---|---|---|---|---|---|
| SVR | - | 0.344 | 0.517 | 0.567 | 0.660 | 0.683 | 0.692 |
| LSTM | - | 0.316 | 0.583 | 0.671 | 0.744 | **0.817** | **0.815** |
| Best HI-model | **0.655** | 0.688 | 0.721 | 0.703 | 0.719 | 0.747 | 0.740 |
| Proposed approach | **0.655** | **0.738** | **0.776** | **0.754** | **0.786** | 0.797 | 0.793 |

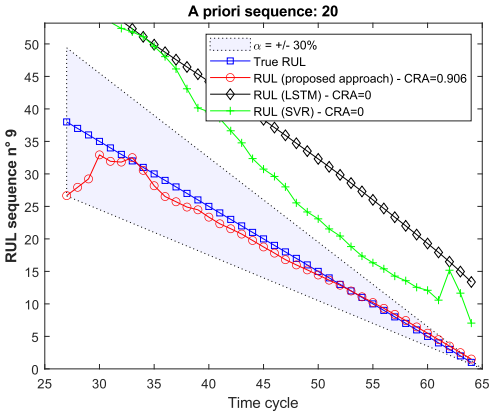
modelling the relationship between the input and the output. It can predict the RUL by mapping the HIs (input) directly to the RUL values (output). Unlike SVR that predicts the RUL directly considering each time point independently, LSTM is a recurrent neural network designed to learn the long term dependencies. It can remember information for long periods of time, it has been applied for RUL estimation of lithium-ion batteries (Y. Zhang et al. 2018), aircraft turbofan engines (Wu et al. 2018), and bearings (B. Zhang et al. 2019). Table 3.11 indicates that the proposed approach yields better results when few run-to-failure sequences are available in advance, comparing to SVR and LSTM. When more a priori degradation sequences are available, LSTM outperforms the proposed approach. Table 3.12 presents comparison results for RUL estimation of faulty bearings with an inner race fault. As in the previous table, it can be seen that the proposed method with the ensemble of models provides better accuracies when few a priori sequences are available. The results confirm that the proposed approach is very efficient for RUL estimation dealing with insufficient a priori sequences.

Figure 3-18 shows a comparison of the predicted RUL of sequence #9 using the proposed approach, SVR, and LSTM. In Figure 3-18a and 3-18b, the RUL of sequence #9 (medium speed) is predicted, by using a priori sequences with different degradation speeds: #1 (very high speed) and #20 (very low speed). Comparing to the RUL prediction using the proposed approach, the RUL predicted using SVR and LSTM is very far from the true RUL. This can be explained by the fact that SVR and LSTM are static models, the models are trained offline on sequences of run-to-failure data, then the estimated models are used online for RUL prediction. Contrary to these methods, the proposed approach is an adaptive approach by using an ensemble of

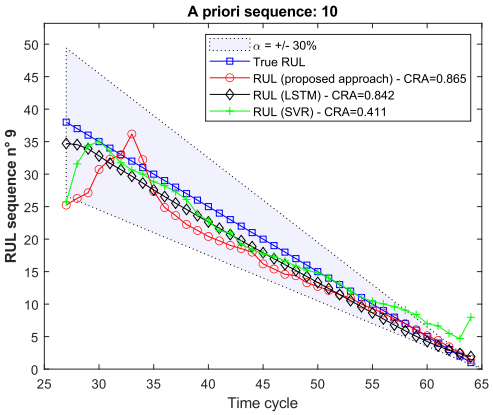
adaptive models. Only the type of models and HIs are selected in the offline phase, while in the online phase, the models are fitted and updated with each new time cycle in order to estimate the RUL. However, when predicting RUL of sequence #9 using as a priori sequence #10 (see Figure 3-18c) with the same degradation speed (medium speed), RUL's prediction accuracy using SVR or LSTM is comparable to the proposed ensemble approach.




(a) Predicted RUL using a priori sequence with different speed (fast)




(b) Predicted RUL using a priori sequence with different speed (slow)




(c) Predicted RUL using a priori sequence with same speed (medium)

Figure 3-18: RUL prediction comparison for the sequence #9 with the proposed approach, SVR, and LSTM

## 3.5 Experimentation using degradation data of an aircraft engine

### 3.5.1 Dataset presentation

The performance of the proposed approach is validated using a public dataset named Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). The dataset represents the damage propagation of an aircraft gas turbine engines. C-MAPSS dataset is generated by NASA (Saxena et al. 2008b), this data have been widely used to compare RUL prediction methods in the literature (Louen et al. 2013; Babu et al. 2016; Al-Dulaimi et al. 2019; C. Zheng et al. 2018; X. Li et al. 2018). The subdataset FD001 is considered for this validation, which includes one type of fault and one mode of operating conditions. This subdataset contains 100 run-to-failure sequences in the training set, and 100 sequences for testing that stop at some time before failure. For each sequence, there are 21 sensors measurements about the system conditions (e.g., temperature, pressure, and rotational speed).

In order to show the performance of our proposed approach, a degradation scenario is considered for this dataset. The 100 sequences in the training set are sorted according to their length. Then, three sequences are selected for the smallest length (number 1, 2 and 3 after sorting), the largest length (number 98, 99 and 100 after sorting), and the medium length (number 49, 50 and 51 after sorting). Hence, a degradation scenario of 9 sequences is obtained with different speeds, as seen in Table 3.13.

### 3.5.2 Results and discussion

#### Data processing and features definition

First, some sensors are not selected from the 21 sensors in the dataset. The sensors # 1, 5, 6, 10, 16, 18, and 19 are discarded because their values remain unchanged during operation, also sensors # 9 and 14 are discarded because of their strong variation. Hence, sensors measurements # 2, 3, 4, 7, 8, 11, 12, 13, 15, 17, 20, and 21 are selected

Table 3.13: Degradation scenario

| Sequence # | Degradation speed | Time cycle length |
|---|---|---|
| 1 | | 128 |
| 2 | Fast | 135 |
| 3 | | 137 |
| 4 | | 198 |
| 5 | Medium | 199 |
| 6 | | 199 |
| 7 | | 336 |
| 8 | Slow | 341 |
| 9 | | 362 |

and presented in Figure 3-19.

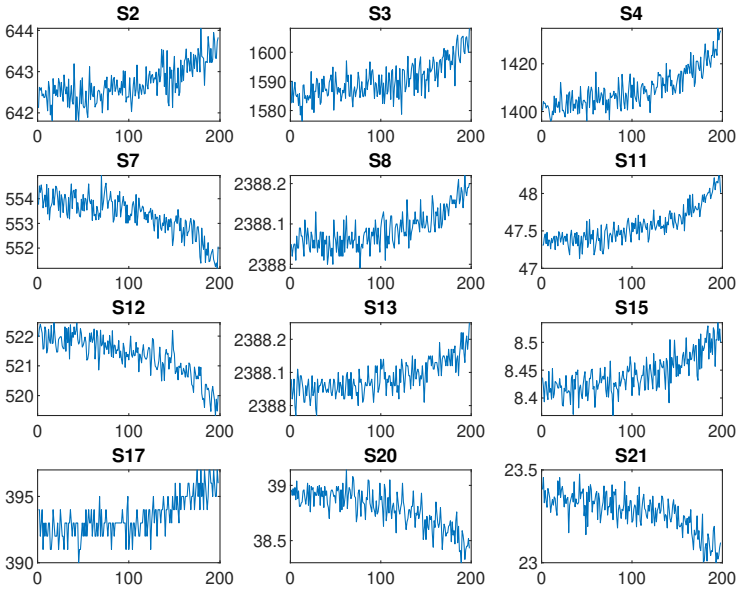



Figure 3-19: Selected sensor values for the sequence #4

Then, a moving mean is applied in order to smooth the sensors' values and to remove the noise, and the sensors that exhibit a negative exponential trend are flipped in order to obtain a positive exponential trend for all the sensor values. The features library (HI library) contains the preprocessed raw measurements of the sensors, and is presented in Table 3.14.

Table 3.14: Features library for the C-MAPSS dataset

| HI index | $HI_1$ | $HI_2$ | $HI_3$ | $HI_4$ | $HI_5$ | $HI_6$ | $HI_7$ | $HI_8$ | $HI_9$ | $HI_{10}$ | $HI_{11}$ | $HI_{12}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sensor | S2 | S3 | S4 | S7 | S8 | S11 | S12 | S13 | S15 | S17 | S20 | S21 |

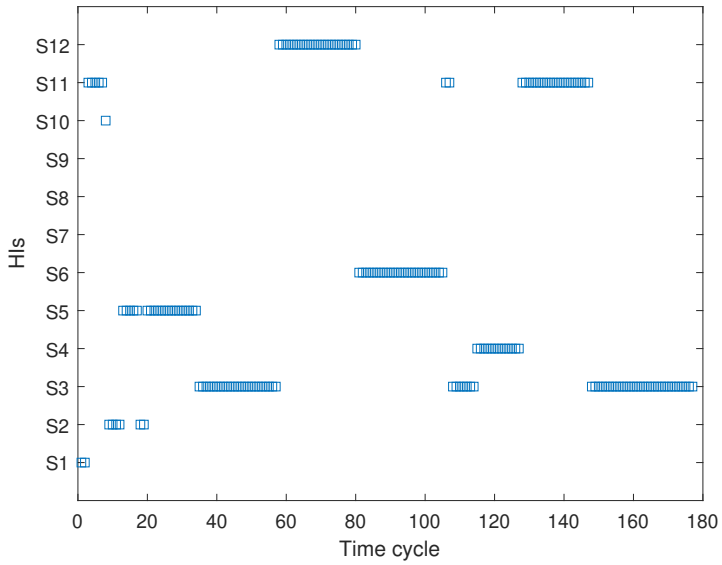
Table 3.15: Degradation scenarios

| Sequence N | Degradation speed | Accuracy (Blind) |
|---|---|---|
| 1 | | 0.502 |
| 2 | Fast | 0.463 |
| 3 | | 0.431 |
| 4 | | 0.608 |
| 5 | Medium | 0.515 |
| 6 | | 0.546 |
| 7 | | 0.668 |
| 8 | Slow | 0.687 |
| 9 | | 0.612 |
| Mean | | 0.559 |

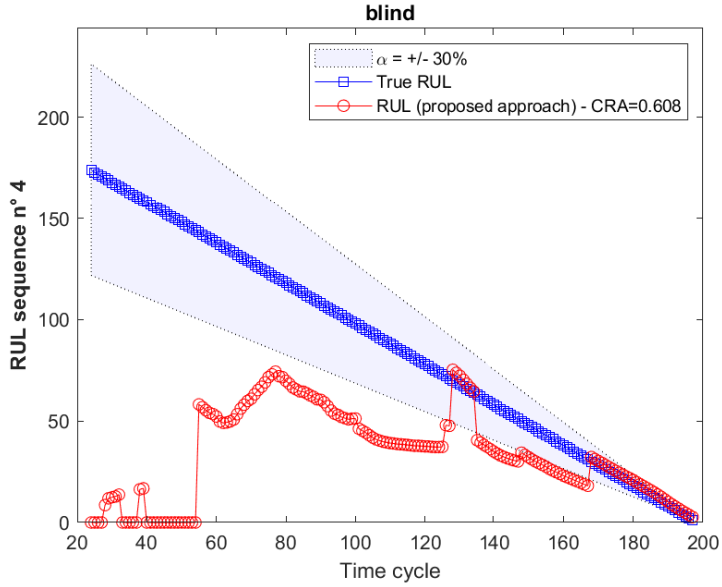
### RUL estimation using the blind path

In this dataset, there is no need to detect the fault, since the degradation sequences are run-to-failure data which mean that the degradation starts with the beginning of the sequence. When no a priori sequences are available, the blind path is triggered. In the blind path, the best HI is selected dynamically among the available HIs in the features library, where the selection is achieved using the selection criterion presented in Eq. (3.6). An example of dynamically selected HIs for sequence #4 is presented in Figure 3-20a, it can be observed that the selected HI changes over time. The corresponding predicted RUL is shown in Figure 3-20b, where it can be seen that the RUL is less accurate at the beginning of the degradation, and then it becomes more precise when approaching the failure. This is thanks to the selection of the best HI by using the dynamic HI selection, also thanks to the adaptive GLM model that can be updated with each new collected data point.

The RUL estimation accuracies for all the sequences using the blind path are presented in Table 3.15. The mean of computed RUL estimation accuracies is around 0.559, this estimation accuracy is suitable, knowing that the prediction is achieved in a blind manner. This results using the blind path will be compared next when having




(a) Online selected HI for sequence #4




(b) RUL estimation using the blind path for sequence #4

Figure 3-20: Example of RUL estimation using the blind path

more available a priori sequences.

### RUL estimation using the informed path

When some a priori sequences are available, the HI-model pairs that give a good CRA accuracy greater than a predefined threshold, are selected. The available HIs

in the features library (Table 3.14) are employed with each of the four extrapolation based models used previously: $\text{Model}_1$ : GLM, $\text{Model}_2$ : quadratic regression, $\text{Model}_3$ : exponential model, and $\text{Model}_4$ : double exponential model. In offline, the HI-model pairs that show better accuracy than the threshold of 0.559 are selected. The threshold is the average of accuracy for the blind path in order to select the HI-model pairs that give better performance than the blind path. The RUL can be predicted using all the HI-model combinations by assigning a weight near to zero to the combinations with low accuracy, since the weights are computed according to the CRA accuracy in an exponential manner (see Eq. (3.13)). However, using all the combinations can lead to high computation time, for this reason, setting a threshold in order to select only the best HI-model pairs is necessary to reduce the computation time. In online, several RULs are estimated for a new incoming sequence using the selected HI-model pairs, while the final RUL is computed using the weighted mean presented in Eq. (3.14).

In order to validate our proposed approach with the informed path, the RUL is predicted for all the nine sequences describing the degradation scenarios in Table 3.13. When only one a priori sequence is available, the RUL is predicted for each sequence as presented in Table 3.16. Table 3.16 presents the CRAs for the RUL estimation of the sequences in the scenario using one a priori sequence. It can be observed that the diagonal CRAs in the table are the highest because the RUL is estimated for each sequence using as a priori the same sequence. When two and three a priori sequences are available, sequences are added randomly to the a priori sequence as shown in Table 3.17 and Table 3.18, respectively. From these tables, it can be seen that the accuracies are slightly improved. The improvement in accuracy when more a priori sequences are collected can be shown by computing the average of each table row (predicted sequence).

Table 3.19 shows the average accuracies of each row and the overall accuracy for each number of a priori sequence. From Table 3.19, the accuracy improvement when using more a priori sequence can be seen in the overall accuracy. Moreover, the standard deviation is computed for each row in order to show the robustness of

Table 3.16: RUL estimation accuracies (CRAs) for the degradation scenarios with one a priori sequence

| | | A priori sequences (1 sequence) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 |
| Predicted sequences | S1 | 0.863 | 0.794 | 0.624 | 0.727 | 0.589 | 0.773 | 0.735 | 0.776 | 0.736 |
| | S2 | 0.796 | 0.812 | 0.698 | 0.717 | 0.616 | 0.787 | 0.741 | 0.715 | 0.659 |
| | S3 | 0.587 | 0.787 | 0.898 | 0.778 | 0.799 | 0.829 | 0.747 | 0.728 | 0.675 |
| | S4 | 0.63 | 0.757 | 0.722 | 0.855 | 0.842 | 0.865 | 0.864 | 0.726 | 0.775 |
| | S5 | 0.505 | 0 | 0.245 | 0.824 | 0.893 | 0.419 | 0.722 | 0.865 | 0.412 |
| | S6 | 0.821 | 0.883 | 0.607 | 0.733 | 0.654 | 0.805 | 0.709 | 0.67 | 0.718 |
| | S7 | 0.645 | 0.631 | 0.598 | 0.845 | 0.791 | 0.809 | 0.918 | 0.789 | 0.859 |
| | S8 | 0.601 | 0.664 | 0.405 | 0.5 | 0.685 | 0.595 | 0.672 | 0.86 | 0.503 |
| | S9 | 0.648 | 0.567 | 0.538 | 0.881 | 0.813 | 0.736 | 0.855 | 0.842 | 0.886 |

Table 3.17: RUL estimation accuracies (CRAs) for the degradation scenarios with two a priori sequences

| | | A priori sequences (2 sequences) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | S3 | S6 | S5 | S7 | S4 | S8 | S9 | S1 | S2 |
| | | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 |
| Predicted sequences | S1 | 0.807 | 0.697 | 0.627 | 0.714 | 0.575 | 0.773 | 0.732 | 0.716 | 0.743 |
| | S2 | 0.782 | 0.79 | 0.588 | 0.692 | 0.617 | 0.734 | 0.685 | 0.792 | 0.712 |
| | S3 | 0.751 | 0.715 | 0.831 | 0.74 | 0.786 | 0.806 | 0.684 | 0.667 | 0.615 |
| | S4 | 0.821 | 0.792 | 0.829 | 0.866 | 0.876 | 0.764 | 0.841 | 0.83 | 0.809 |
| | S5 | 0.621 | 0.38 | 0.875 | 0.843 | 0.882 | 0.816 | 0.692 | 0.825 | 0.655 |
| | S6 | 0.717 | 0.778 | 0.669 | 0.682 | 0.647 | 0.718 | 0.705 | 0.675 | 0.718 |
| | S7 | 0.775 | 0.845 | 0.849 | 0.907 | 0.841 | 0.829 | 0.903 | 0.819 | 0.844 |
| | S8 | 0.691 | 0.735 | 0.704 | 0.648 | 0.651 | 0.848 | 0.575 | 0.862 | 0.616 |
| | S9 | 0.817 | 0.743 | 0.862 | 0.839 | 0.804 | 0.845 | 0.874 | 0.852 | 0.844 |

Table 3.18: RUL estimation accuracies (CRAs) for the degradation scenarios with three a priori sequences

| | | A priori sequences (3 sequences) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | S2 | S7 | S4 | S8 | S1 | S3 | S5 | S9 | S6 |
| | | S3 | S6 | S5 | S7 | S4 | S8 | S9 | S1 | S2 |
| | | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 |
| Predicted sequences | S1 | 0.839 | 0.754 | 0.615 | 0.734 | 0.697 | 0.736 | 0.678 | 0.715 | 0.716 |
| | S2 | 0.796 | 0.772 | 0.631 | 0.76 | 0.724 | 0.689 | 0.665 | 0.729 | 0.723 |
| | S3 | 0.775 | 0.696 | 0.813 | 0.73 | 0.704 | 0.777 | 0.796 | 0.675 | 0.625 |
| | S4 | 0.804 | 0.786 | 0.881 | 0.81 | 0.85 | 0.693 | 0.849 | 0.762 | 0.802 |
| | S5 | 0.261 | 0.519 | 0.874 | 0.754 | 0.853 | 0.815 | 0.863 | 0.85 | 0.657 |
| | S6 | 0.793 | 0.735 | 0.65 | 0.669 | 0.678 | 0.731 | 0.699 | 0.668 | 0.716 |
| | S7 | 0.638 | 0.879 | 0.87 | 0.891 | 0.854 | 0.809 | 0.87 | 0.798 | 0.839 |
| | S8 | 0.682 | 0.721 | 0.685 | 0.82 | 0.772 | 0.851 | 0.754 | 0.872 | 0.713 |
| | S9 | 0.743 | 0.793 | 0.833 | 0.852 | 0.851 | 0.879 | 0.858 | 0.869 | 0.849 |

Table 3.19: Average of RUL estimation accuracies (CRAs) with different number of a priori sequences

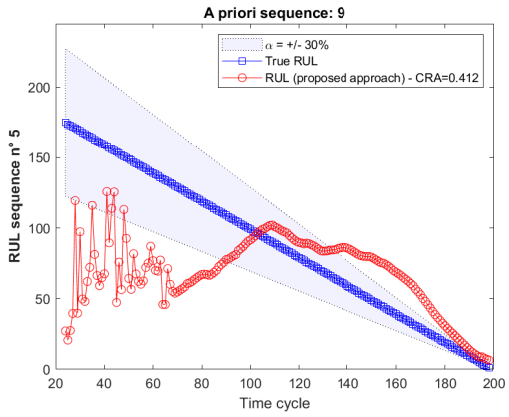
| | | Number of a priori sequences | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | CRA (1seq) | CRA (2seq) | CRA (3seq) | CRA (4seq) | CRA (5seq) | CRA (6seq) | CRA (7seq) | CRA (8seq) | CRA (9seq) |
| Predicted sequences | S1 | 0.735+0.0842 | 0.709+0.0710 | 0.720+0.0604 | 0.722+0.0424 | 0.740+0.0371 | 0.745+0.0257 | 0.753+0.0164 | 0.752+0.0200 | 0.750+0 |
| | S2 | 0.727+0.0652 | 0.710+0.0738 | 0.721+0.0526 | 0.72+0.0328 | 0.714+0.0304 | 0.728+0.0293 | 0.734+0.0142 | 0.737+0.0195 | 0.747+0 |
| | S3 | 0.759+0.0898 | 0.733+0.0700 | 0.732+0.0625 | 0.703+0.0648 | 0.706+0.0736 | 0.695+0.0638 | 0.694+0.0397 | 0.696+0.0278 | 0.698+0 |
| | S4 | 0.782+0.0815 | 0.825+0.0347 | 0.804+0.0555 | 0.811+0.0412 | 0.819+0.0357 | 0.813+0.0292 | 0.817+0.0170 | 0.817+0.0143 | 0.818+0 |
| | S5 | 0.543+0.3070 | 0.732+0.1640 | 0.716+0.2070 | 0.818+0.0577 | 0.842+0.0290 | 0.844+0.0202 | 0.846+0.0103 | 0.844+0.0099 | 0.843+0 |
| | S6 | 0.733+0.0882 | 0.701+0.0383 | 0.704+0.0447 | 0.688+0.0157 | 0.687+0.0129 | 0.687+0.0185 | 0.688+0.0153 | 0.688+0.0094 | 0.685+0 |
| | S7 | 0.765+0.1130 | 0.846+0.0404 | 0.828+0.0778 | 0.845+0.0315 | 0.852+0.0284 | 0.846+0.0340 | 0.846+0.0240 | 0.846+0.0159 | 0.843+0 |
| | S8 | 0.610+0.1330 | 0.703+0.0981 | 0.763+0.0705 | 0.792+0.0657 | 0.801+0.0544 | 0.806+0.0411 | 0.814+0.0376 | 0.822+0.0309 | 0.838+0 |
| | S9 | 0.752+0.1360 | 0.831+0.0394 | 0.836+0.0426 | 0.855+0.0130 | 0.858+0.0108 | 0.856+0.0119 | 0.857+0.0087 | 0.859+0.0075 | 0.857+0 |
| **Overall accuracy** | | **0.712+0.1220** | **0.755+0.0699** | **0.758+0.0749** | **0.773+0.0405** | **0.780+0.0347** | **0.780+0.0304** | **0.783+0.0204** | **0.784+0.0173** | **0.787+0** |

prediction. It can be seen that the overall standard deviation is decreasing when collecting more a priori sequences. Hence, our proposed approach can improve the accuracy of prediction as well as the robustness when more a priori sequences are available. The RUL estimation accuracy improvement is illustrated in Figure 3-21, it can be observed that when collecting more a priori sequences, the RUL estimation is improved. This is thanks to the proposed HI-model pairs selection that can update the selected pairs in offline for each new stored sequence.

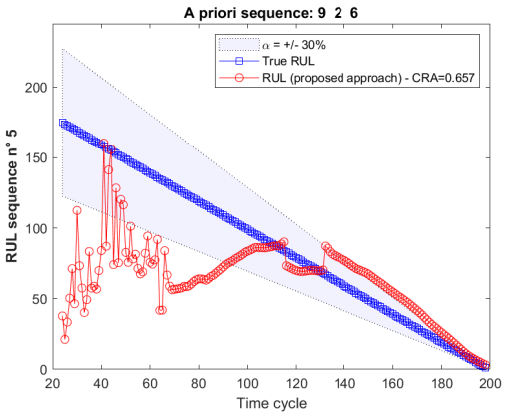
Table 3.20: Comparison of the RUL estimation overall accuracies

| | CRA (Blind) | CRA (1seq) | CRA (2seq) | CRA (3seq) | CRA (4seq) | CRA (5seq) | CRA (6seq) | CRA (7seq) | CRA (8seq) | CRA (9seq) |
|---|---|---|---|---|---|---|---|---|---|---|
| SVR | - | 0.167 | 0.325 | 0.408 | 0.498 | 0.559 | 0.620 | 0.663 | 0.702 | 0.742 |
| LSTM | - | 0.362 | 0.448 | 0.540 | 0.615 | 0.657 | 0.749 | **0.794** | **0.851** | **0.912** |
| Best HI-model | **0.559** | 0.487 | 0.635 | 0.586 | 0.647 | 0.648 | 0.659 | 0.661 | 0.670 | 0.676 |
| Proposed approach | **0.559** | **0.712** | **0.755** | **0.758** | **0.773** | **0.780** | **0.780** | 0.783 | 0.784 | 0.787 |

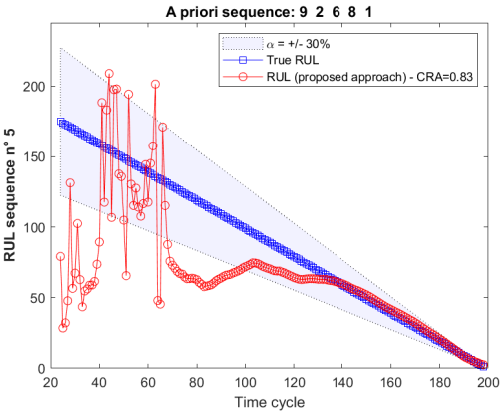
Table 3.20 compares the overall accuracies of the RUL estimation using the proposed approach, which uses a set of best HI-models, the top best HI-model, SVR, and LSTM. SVR and LSTM are machine learning techniques that have shown their effectiveness for the RUL estimation (Benkedjouh et al. 2013; Soualhi et al. 2014; Y. Zhang et al. 2018; Wu et al. 2018). The proposed approach combines several HI-model pairs for RUL estimation, and it is compared when using only the best HI-model pair. Based on Table 3.20, it can be observed that the proposed approach outperforms the other techniques when few a priori sequences are available, this is thanks to the adaptability of our proposed approach and also to the combination of several HI-model pairs. However, when several a priori sequences are available in



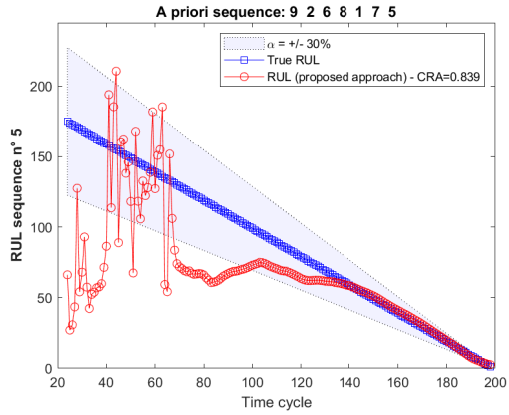

(a) Predicted RUL using a priori sequence #9




(b) Predicted RUL using a priori sequence #9, #2, and #6




(c) Predicted RUL using a priori sequence #9, #2, #6, #8, and #1

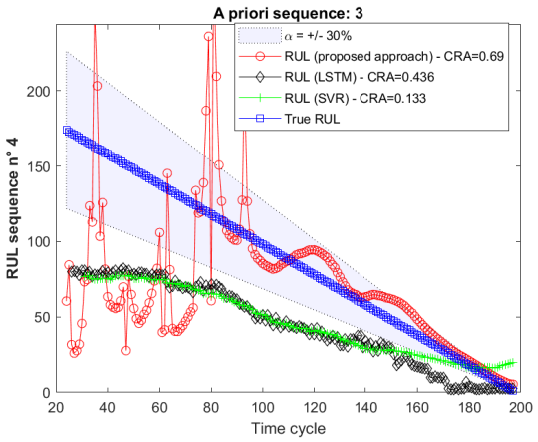



(d) Predicted RUL using a priori sequence #9, #2, #6, #8, #1, #7, and #5

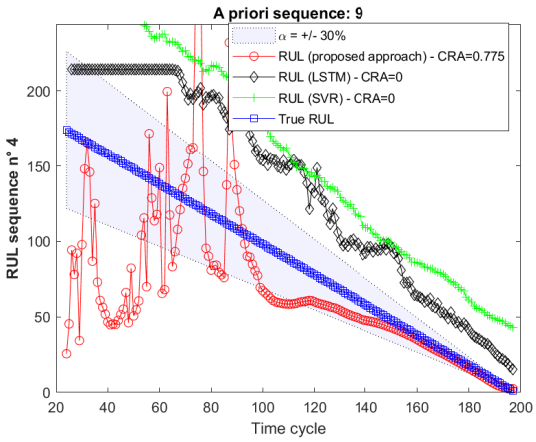
Figure 3-21: RUL prediction for sequence #5 using different available a priori sequences

offline (more than 6 sequences), the RUL estimation accuracy using our approach converges while the LSTM outperforms the proposed approach.

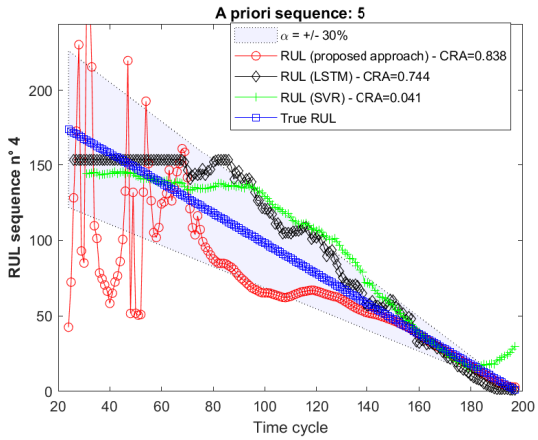
Figure 3-22 shows the adaptability of our proposed approach compared with SVR and LSTM methods. The RUL is estimated for sequence #4 which is a sequence with a medium degradation speed, the RUL is estimated using as a priori one sequence with different degradation speeds (i.e., fast, medium, and slow). It can be observed that when the RUL is predicted using a priori sequence with a degradation speed different from the predicted one, our approach has a good accuracy comparing with SVR and LSTM that show a low accuracy. Our proposed approach can estimate the




(a) Predicted RUL using a priori sequence with different speed (fast)




(b) Predicted RUL using a priori sequence with different speed (slow)




(c) Predicted RUL using a priori sequence with same speed (medium)

Figure 3-22: RUL prediction for sequence #4 using a priori sequences with different degradation speed

RUL with good accuracy even when the degradation speed of the a priori sequence is different from the predicted one. This is thanks to the adaptability of our proposed approach that combines different adaptive models. When the degradation speed of the a priori sequence is similar to the new sequence, it can be seen that the LSTM shows good accuracy for RUL estimation, SVR shows a low accuracy because CRA accuracy penalizes more the errors of prediction near to the failure.

## 3.6 Conclusion

In this chapter, a data-driven approach is proposed for RUL estimation dealing with insufficient a priori run-to-failure sequences. Firstly, a blind case is considered, where no a priori sequences are available, thus no a priori knowledge is available allowing to guide the choice of the most suitable or efficient health indicators (HIs) to use. To overcome this issue, the best HI is selected dynamically using a selection criterion for each time cycle, this criterion indicates the degree to which a HI fits an extrapolation model. After collecting some a priori run-to-failure sequences offline, the informed path is triggered. In this path, the RUL is predicted using an ensemble of models fitted with the corresponding HIs, where the final RUL is predicted using a weighted mean. The HI-model pairs and their corresponding weights are determined offline.

The blind path of the proposed approach is validated using real vibration data collected from a degraded shaft of a wind turbine. It showed promising results compared to the use of predefined HIs. The blind and the informed paths of the proposed approach are validated using degradation data of a rolling bearing and a turbofan engine of an aircraft. In the degradation scenarios of these datasets, there are several sequences with different degradation speeds. The proposed approach showed several advantages. First, it exploits the diversity of the different models and HIs allowing the improvement the RUL's prediction accuracy. Second, the robustness of RUL prediction for the new incoming sequences increases over time thanks to update of the selected HI-model pairs by integrating the collected degradation data. This can be seen through the decrease of the variation (STD) of the RUL's predictions over time. Third, the proposed approach showed better RUL prediction accuracy compared to the well-known methods applied for RUL prediction: support vector machine and long short term memory in particular when there are few available degradation data. Finally, the proposed method can be adaptive to the different degradation speeds because it uses an ensemble of different online adaptive models.

As demonstrated through the results of this chapter, RUL estimation using LSTM outperformed our proposed approach when collecting an important number of a priori

run-to-failure sequences. For this reason, a new data-driven approach based on the use of deep learning models is proposed in the next chapter, which deals with the case where multiple run-to-failure sequences are available.

# Chapter 4

# Prognostic approach with multiple a priori degradation sequences

## 4.1 Introduction

As we have seen in chapter 3, when there is no or few insufficient a priori run-to-failure degradation sequences, the blind and the informed paths, respectively, are triggered for the RUL estimation. In this case, the RUL is predicted in an indirect way, where the HI is computed and extrapolated until reaching the failure threshold. The indirect RUL estimation is efficient when there is insufficient a priori run-to-failure sequences. On the other hand, when an important number of a priori sequences are available, employing direct RUL estimation is suitable for efficient RUL estimation. Direct RUL estimation has several advantages. Firstly, it is not necessary to understand the operation of the system and its different operating modes, most of the data-driven techniques applied to the direct RUL estimation (e.g., ANN) can handle the variation of the operating modes in complex dynamic systems. Secondly, it is not necessary to extract and select features in order to build a suitable health indicator for extrapolation, the direct RUL estimation way maps the raw data collected from sensors directly to the RUL. Finally, it is not necessary to predefine a failure threshold for the RUL estimation, knowing that defining a failure threshold is challenging and requires domain expertise.

This chapter focuses on the case where multiple a priori run-to-failure sequences are available. A new data-driven approach for direct RUL estimation is proposed based on a deep ensemble method. The proposed approach combines the decisions of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) models, which are kn won to be efficient methods for RUL estimation when several a priori run-to-failure sequences are available. The proposed approach aims to improve the reliability as well as the accuracy of the RUL prediction by exploiting the diversity of CNN and LSTM models. It is validated using two datasets: the first dataset is C-MAPSS used in chapter 3, which are degradation data collected from an aircraft engine, the data are generated by the National Aeronautics and Space Administration (NASA) and named C-MAPSS dataset (Saxena et al. 2008b). The second dataset comprises degradation data collected from sensors in order to estimate the RUL of an industrial filter system before clogging. These data are provided by the fifth European PHM Society conference for the data challenge competition.

The chapter is organized as follows: section 4.2 presents the proposed approach for RUL estimation. The experimentation and obtained results on the degraded aircraft engine and the clogged filter datasets are presented sections 4.3 and 4.4, respectively. Finally, section 4.5 ends the chapter with concluding remarks.

## 4.2 Proposed approach

The proposed approach, illustrated in Figure 4-1, includes two phases: an offline phase for tuning and training the models and an online phase for estimating the RUL from new incoming observations. In the offline phase (training phase), the historical run-to-failure sequences are first preprocessed by selecting the significant input data (sensors), normalizing, segmenting them into windows, and setting the true RUL of these sequences (labels). After data preprocessing, different CNN and LSTM models are trained in order to select the optimal hyperparameters by using k-fold cross-validation. Then, the fusion weights are computed according to the performance of the validation data. In the online phase (testing phase), the data are preprocessed as

in the offline phase. Then, the RULs are predicted using CNN and LSTM models, while the predicted RULs by each model are then merged using a weighted mean in order to obtain the final RUL. In the next subsections, the data processing step is explained, the CNN and LSTM models are described, and the decision fusion step is presented.

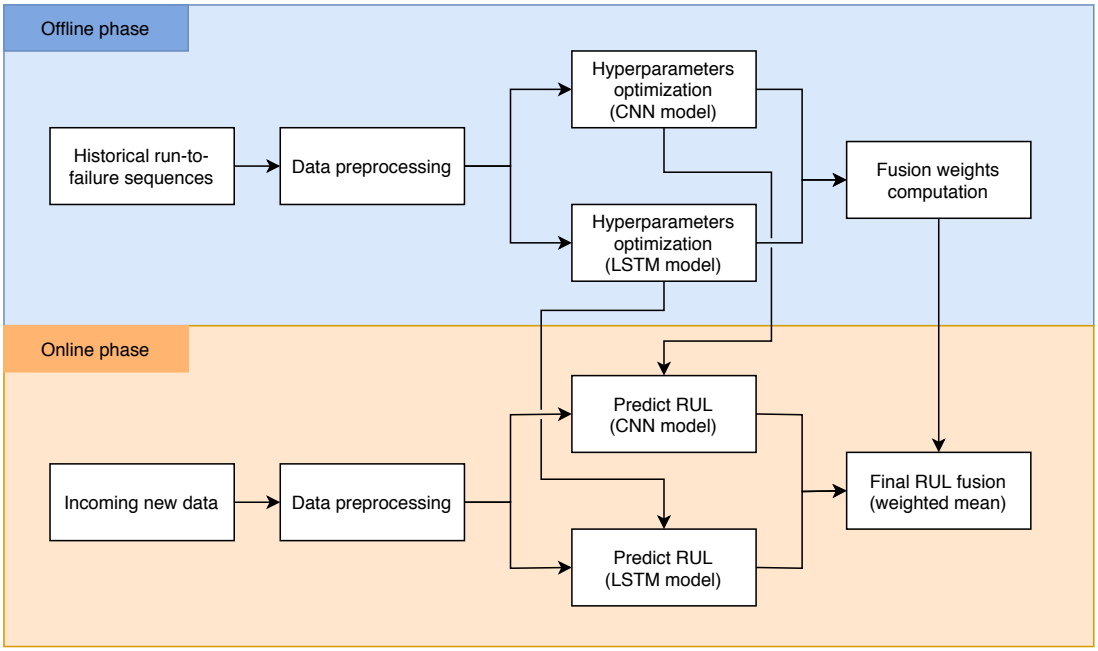



Figure 4-1: Proposed deep ensemble approach for RUL estimation

### 4.2.1 Data processing

When the data collected from sensors are acquired, they are first preprocessed before starting the RUL estimation. The collected raw data from sensors are in different ranges, this may lead to unequal weight computation in the deep neural network (CNN and LSTM). Hence, the data are first normalized using z-score normalization, which is computed as follow:

$$Norm\left(x_s\right) = \frac{x_s - \mu_s}{\sigma_s} \tag{4.1}$$

where $Norm\left(x_s\right)$ represents the normalized values, $x_s$ are the values of the sensor $s$, $\mu_s$ is the mean and $\sigma_s$ is the standard deviation of each sensor $s$.

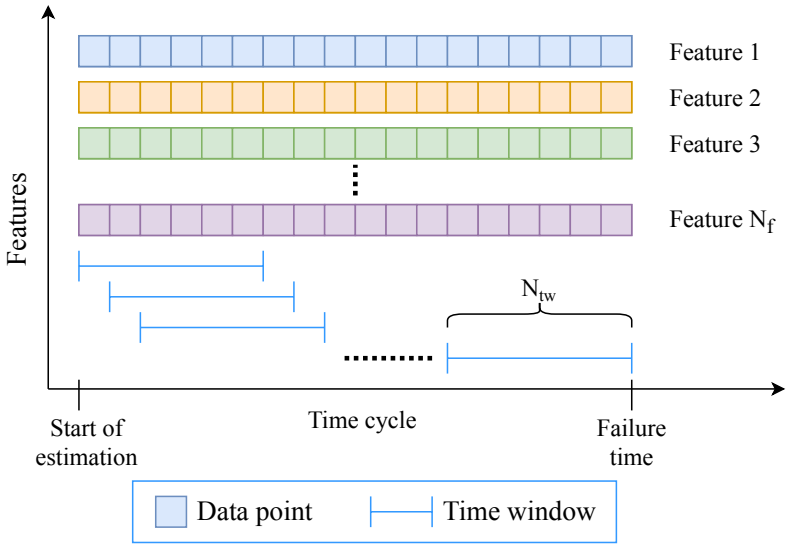



Figure 4-2: Sliding time window used as input

The normalized input data are then segmented using a time window as illustrated in Figure 4-2, where the multiple sensors measurements with length $N_f$ are shown as different features. A fixed length sliding time window of length $N_{tw}$ is employed for the segmentation of the consecutive data points. Hence, a two dimensional input is obtained for each time cycle ($N_{tw} \times N_f$). In the training phase, this 2D input is fed to the deep learning models while the output of the models is the true RUL.

### 4.2.2 RUL estimation using convolutional neural network

The convolutional neural network is developed mainly for computer vision by LeCun et al. (1995), it is efficient for automatic feature extraction. The adopted CNN model is a 1-dimensional CNN (1D CNN) that can handle time series signals. CNN consists of several consecutive convolution layers for features extraction. In the convolution layer, several filters are convolved with the input data in order to generate the features map (as illustrated in Figure 4-3).

The convolution layer operation is represented as follow:

$$f = \phi(U * k + b) \tag{4.2}$$

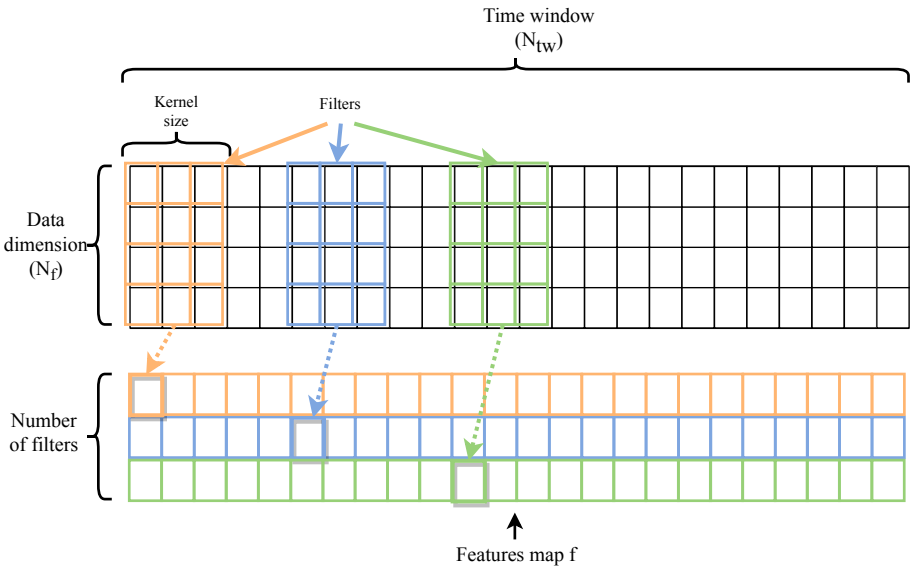



Figure 4-3: Illustration of 1D CNN operation

$k$ represents the convolution filter, $U$ is the input data, and $*$ is the convolution operator, while $b$ is the bias term and $\phi$ represents a nonlinear activation function. $f$ is the obtained features map which represents the learned features by sliding the multiple filters on the input data. Deep CNN architecture has proven its efficiency for RUL estimation (X. Li et al. 2018). The more the network is deep, the more the model can learn high level representation of features. In the proposed CNN architecture, three convolution layers are stacked for efficient features extraction. When the features map is obtained, it is flattened into 1-dimensional shape, and fed to a fully connected layer for RUL prediction as illustrated in Figure 4-4.

### 4.2.3 RUL estimation using long short term memory

Long short term memory is an advanced type of recurrent neural network (Hochreiter et al. 1997), which has been successfully applied for speech recognition and natural language processing. It is able to address the problem of capturing long term memory. LSTM unit structure is shown in Figure 4-5. LSTM unit is composed of the hidden state $h_t$, the cell state $c_t$, and the three gates: forget gate, update gate, and output gate that controls the flow of information in the unit. The forget gate $f_t$ controls the information removal from the previous cell state $c_{t-1}$. The update gate $u_t$ controls

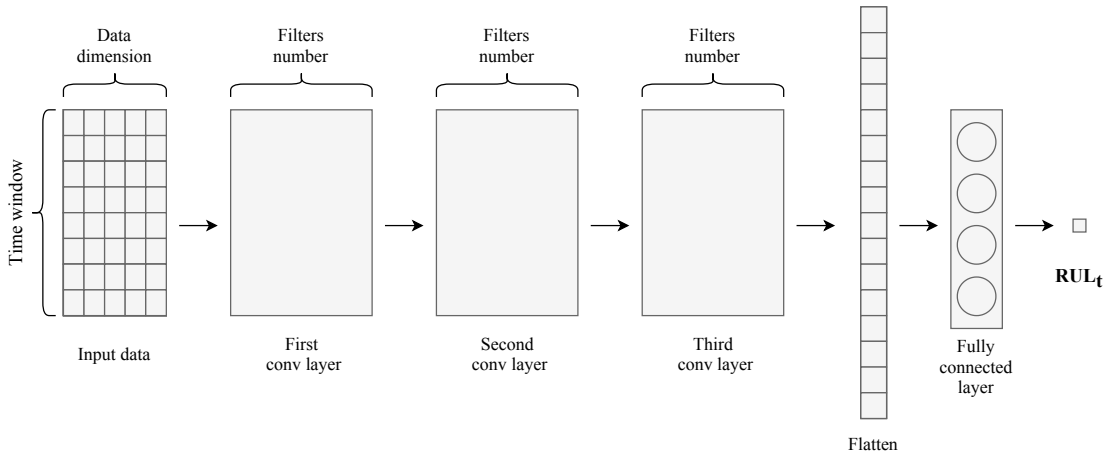



Figure 4-4: Proposed CNN architecture for RUL estimation

the information addition to the current cell state $c_t$. The output gate $o_t$ controls the information that will be carried to the current hidden state $h_t$.

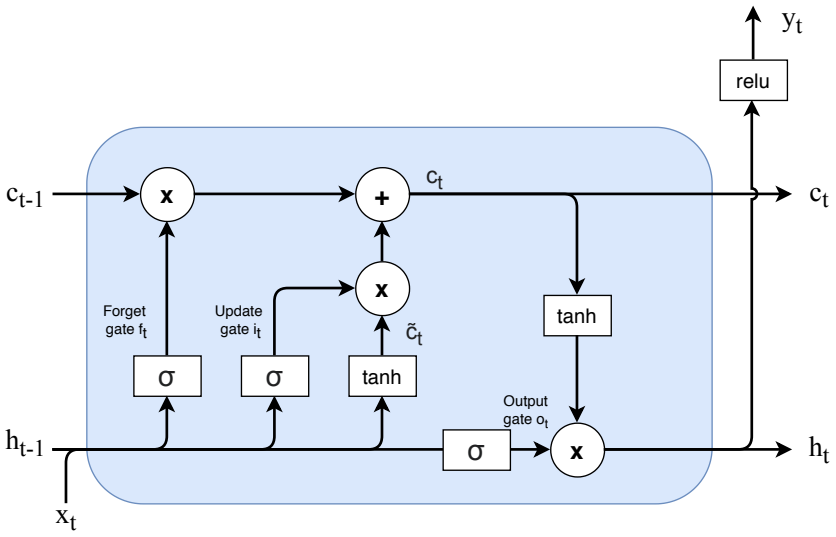



Figure 4-5: Diagram of LSTM cell

The corresponding equations for the explained LSTM unit are presented as follows:

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f) \tag{4.3}$$

$$u_t = \sigma(W_u[x_t, h_{t-1}] + b_u) \tag{4.4}$$

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o) \tag{4.5}$$

$$\tilde{c}_t = tanh(W_c[x_t, h_{t-1}] + b_c) \tag{4.6}$$

$$c_t = f_t * c_{t-1} + u_t * \tilde{c}_t \tag{4.7}$$

$$h_t = o_t * tanh(c_t) \tag{4.8}$$

Where $W_f$, $W_u$, $W_o$, and $W_c$ represent the corresponding weights to the forget gate, update gate, output gate, and the cell state, respectively, while $b_f$, $b_u$, $b_o$, and $b_c$ are their corresponding bias. The weights and bias are estimated during the training phase. $\tilde{c}$ is the candidate cell state, $\sigma$ represents the sigmoid activation function, while $tanh$ is the hyperbolic tangent activation function.

The proposed LSTM architecture is presented in Figure 4-6. It is a many to one architecture where three LSTM layers are stacked in order to discover the underlying patterns embedded in time series. $x_{t1}, x_{t2}, x_{t3}..., x_{tn}$ are the input data points ($t1$ is the first time index of the segmented window while $tn$ is the last index of the input window) and $RUL_{tn}$ is the predicted RUL for the input window.

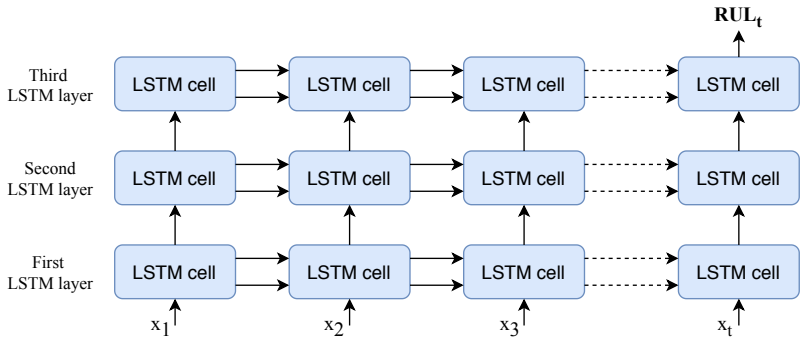



Figure 4-6: Proposed LSTM architecture for RUL estimation

### 4.2.4 Remaining useful life fusion

In the online phase, the predicted RULs using the previously described CNN and LSTM models are then aggregated using the weighted mean as illustrated in Figure 4-7.

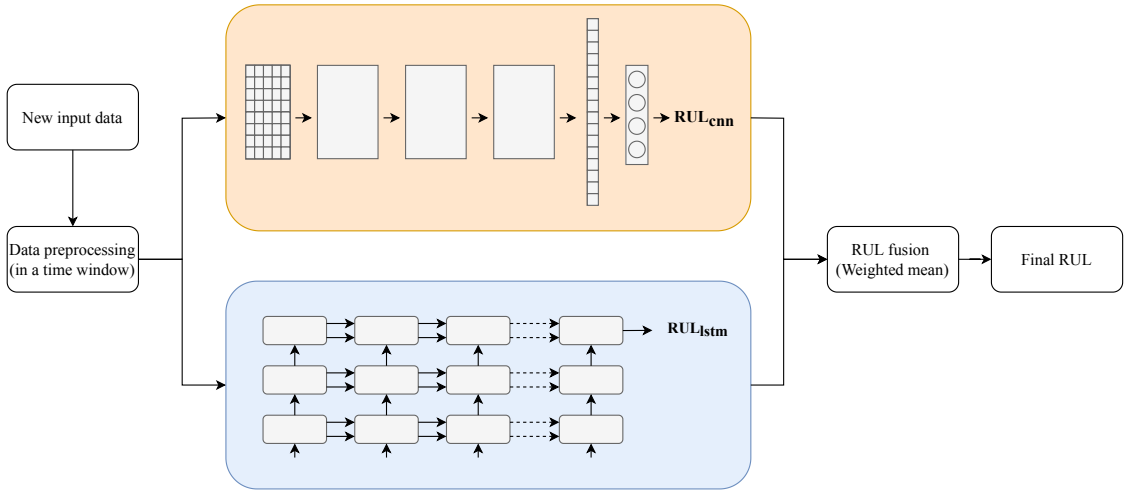



Figure 4-7: Illustration of RUL fusion for a new input data

The weighted mean is applied in (Xia et al. 2020) for aggregating the predicted RUL with different time windows. However, the weights are computed according to the training errors, this may increase the weights of overfitted models (when the training error is small whereas the test error is high). For this reason, in our proposed ensemble approach, the weights are computed according to a validation error. This is done by using the majority of training sequences to train the models, and the remaining sequences are used as validation data for calculating the validation error. The merged RUL is computed using the following equations:

$$W_k = \frac{\frac{1}{ErrVal_k}}{\sum_{k=1}^{n_k} \frac{1}{ErrVal_k}} \tag{4.9}$$

$$rul(t) = \sum_{k=1}^{n_k} W_k . rul(t)_k \tag{4.10}$$

Where $rul(t)$ is the final RUL estimated at each time cycle $t$, $rul(t)_k$ is the RUL estimated by the model $k$ at each time cycle, $n_k$ is the number of models (here $n_k = 2$)

and $W_k$ is the corresponding weight to each model. $ErrVal_k$ represents the validation RMSE errors for each model $k$.

A dropout is applied after the last layer of CNN and LSTM (Srivastava et al. 2014), which is a powerful regularization technique that randomly discards a subset of neurons and their connections during training, it is applied to reduce data overfitting when training deep learning models in order to enhance the model generalization. The dropout probability is set to 0.5, which is an optimal probability value for a wide range of networks and tasks (Srivastava et al. 2014).

## 4.3 Experimentation using data of a degraded aircraft engine

### 4.3.1 Dataset presentation

The performance of the proposed approach is evaluated using the benchmark dataset named Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). The dataset represents the damage propagation of the aircraft gas turbine engines. C-MAPSS dataset is generated by NASA (Saxena et al. 2008b), this data has been widely used to compare RUL prediction methods in the literature (Louen et al. 2013; Babu et al. 2016; Al-Dulaimi et al. 2019; C. Zheng et al. 2018; X. Li et al. 2018).

Table 4.1: C-MAPSS sub-datasets

| Sub-datasets | FD001 | FD002 | FD003 | FD004 |
|---|---|---|---|---|
| Training sequences | 100 | 260 | 100 | 249 |
| Testing sequences | 100 | 259 | 100 | 248 |
| Operating conditions | 1 | 6 | 1 | 6 |
| Fault conditions | 1 | 2 | 2 | 2 |

This dataset is divided into four cases (or sub-datasets), where each case includes several run-to-failure sequences (or trajectories) for training and for testing. The first and third sub-datasets (FD001 and FD003) are generated under one operating condition, FD001 includes one type of fault, while FD003 includes two fault types. The

Table 4.2: C-MAPSS Data format

| Column n# | 1 | 2 | 3-5 | 6-26 |
|---|---|---|---|---|
| Information | Unit index | Time cycle index | Operating conditions | Sensor values |

second and fourth cases (FD002 and FD004) are generated under six operating conditions (variation of three flight condition parameters: aircraft altitude, environmental temperature, and aircraft speed). Table 4.1 summarizes the C-MAPSS sub-datasets properties. The data format contains columns about: unit index, time cycle, three operating conditions or flight conditions, and 21 sensors measurements about the system conditions (e.g., temperature, pressure, and rotational speed), the data format is shown in Table 4.2.

The training dataset contains run-to-failure sequences from healthy to failure, while the test data contains sequences that stop at some time before failure. The goal is to estimate the RUL of the test data until failure. Then, the predicted RUL should be evaluated for all the engine units according to the true RUL which is provided in the datasets. Two evaluation criteria are employed: RMSE (see Eq. (2.7)) and the value of a scoring function defined in (Saxena et al. 2008b). The scoring function penalizes more the overestimated RUL (if the predicted RUL is greater than the true RUL), it is calculated as follows:

$$Score = \begin{cases} \sum_{u=1}^{n_u} \left( e^{\frac{-d(u)}{13}} - 1 \right) & \text{for} \quad d(u) < 0 \\ \sum_{u=1}^{n_u} \left( e^{\frac{d(u)}{10}} - 1 \right) & \text{for} \quad d(u) \geq 0 \end{cases} \tag{4.11}$$

where $n_u$ is the total number of engine units in the test data, $u$ is the engine unit's index. $d(u)$ is the difference between the predicted and true RUL ($rul(u) - true\_rul(u)$).

### 4.3.2 Results and discussion

In the training phase, the data are preprocessed before modeling. Firstly, the data collected by sensors n# 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21 are selected ($N_f = 14$), because the values of the discarded sensors remain unchanged during

operation (as shown in Figure 4-8).

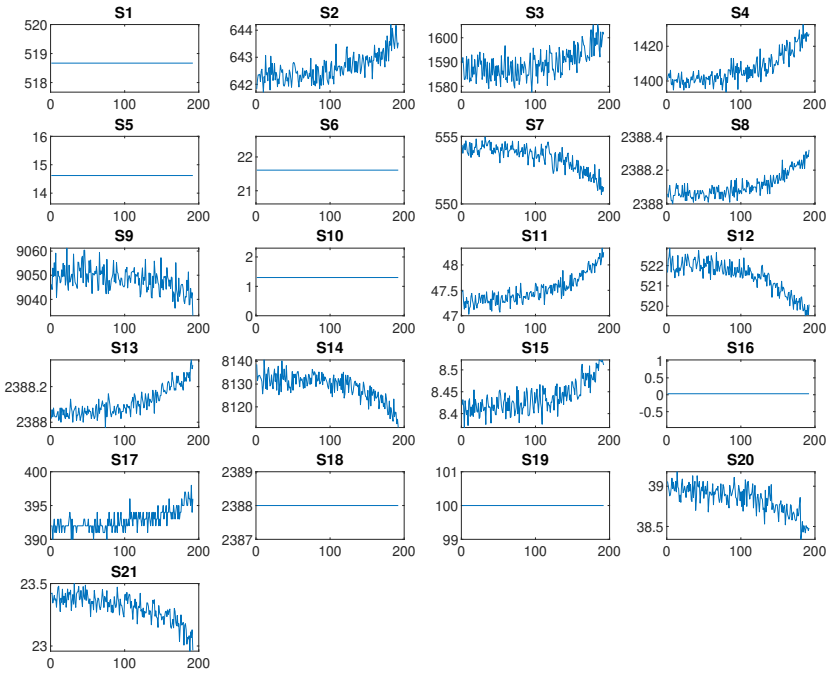



Figure 4-8: Sensors measurement in FD001 sub-datasets

In addition to the selected sensors, the three operating condition (flight condition) measurements are also selected for the cases FD002 and FD004 due to the variation of the operating condition modes as illustrated in Figure 4-9.

In Figure 4-9, operating condition 1 refers to the altitude from sea level ($10^3$ ft), operating condition 2 represents the mach number (a ratio of flow velocity to the speed of sound), and operating condition 3 is the sea-level temperature (°F). For FD001 and FD003 there is only one mode of operating condition represented by the red circle, where for FD002 and FD004 there are six different modes of operating conditions represented by red and blue circles. This variation of operating condition modes results in a variation of the sensor values, which may hide the observation of the system degradation with time. Figure 4-10a represents the values of sensor number #2 in the sub-datasets FD001, the degradation can be observed on the sensor values which is monotonic and increase with time because there is only one mode of

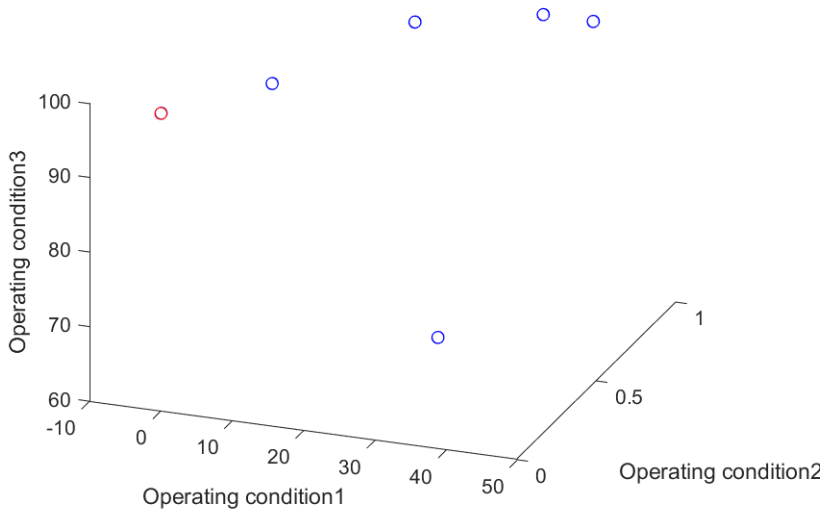
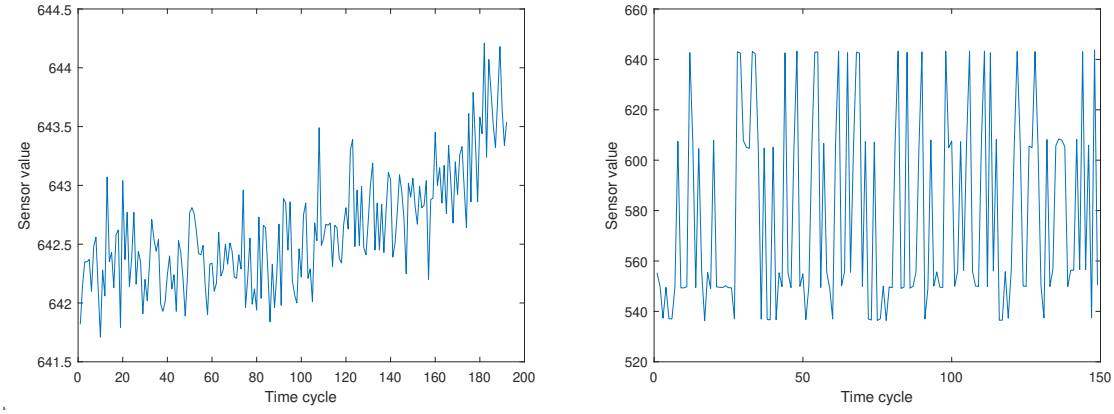



Figure 4-9: Different Operating condition modes for the C-MAPSS dataset

operating conditions. However, in Figure 4-10b which represents the values of sensor number #2 in the sub-datasets FD002, the degradation evolution over time cannot be observed due to the variation of the operating conditions, this may increase the difficulty of RUL estimation.




(a) Values of sensor #2 in FD001 (b) Values of sensor #2 in FD002

Figure 4-10: Sensor #2 values under one (a) and six operating condition modes (b)

In general, the target RUL (true RUL) of a training set sequence should be inversely proportional to the time cycle. For the C-MAPSS datasets, a piece-wise linear

function is proposed by (Heimes 2008) to rectify the training labels. When the training cycle is smaller than a predefined threshold, the true RUL is constant, and the system is considered healthy in this case. Then, when the training cycle is greater than the threshold, the target RUL starts decreasing (as illustrated in Figure 4-11). According to the literature, the threshold is set to 125. This rectification is set because the RUL should not decrease at the beginning of the degradation since the system is always considered under healthy conditions. Also, this rectification will prevent overestimating the RUL prediction (the evaluation score will be larger). We adopted this rectification with the same setting in order to allow a suitable comparison in the same conditions with the related works.

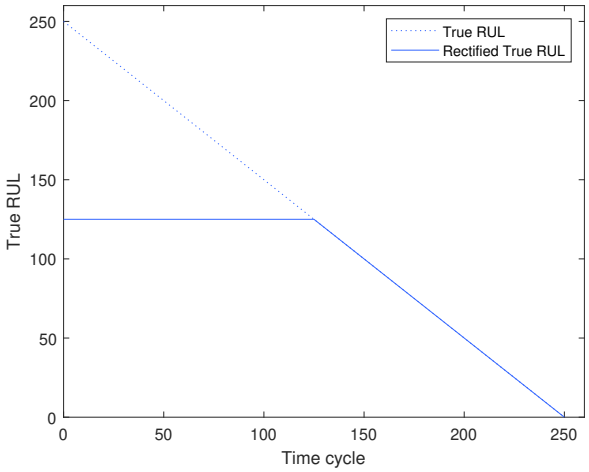



Figure 4-11: Rectified true RUL

The input of the model is constructed using the sliding window, the length of the time window ($N_{tw}$) should be large enough in order to include the maximum of information. On the other side, the minimum running cycle length of the sequences in the test data is 31 cycles, therefore the window length should be less than 31 cycles. Time window of length 15 and 30 cycles are compared for RUL prediction accuracy in (Al-Dulaimi et al. 2019), where the time window with a length of 30 cycles showed better performance, hence for this work, the selected sensor values are normalized using z-score normalization and then segmented using a time window with the length of $N_{tw} = 30$ time cycles.

For training, the RMSE is used as cost function, back-propagation learning is utilized for the updates of the weights in the network using mini-batches. Adam optimizer algorithm (Kingma et al. 2014) is used for training the two models. A kernel size of 10 is adopted which is applied in (X. Li et al. 2018) and showed a good performance on C-MAPSS data. An early stopping criterion is applied in order to stop the training of the network. The training is stopped when the validation error is not decreasing within 10 epochs (iterations). 5-fold cross validation is applied for each model in order to choose the best hyperparameters (mini-batch size $Bs$, learning rate $lr$, filters number $F$ for CNN, and number of hidden units $L$ in LSTM).

Table 4.3 presents the average RMSE errors of the cross validation for the FD001 and FD003, while Table 4.4 presents the average RMSE errors of the cross validation for the FD002 and FD004 (when having multiple operating condition modes). The selected hyperparameters for each subdataset according to the average errors can be seen in bold font.

Table 4.3: Hyperparameters selection for FD001 and FD003

| CNN hyperparameters | FD001 RMSE | FD003 RMSE | LSTM hyperparameters | FD001 RMSE | FD003 RMSE |
|---|---|---|---|---|---|
| **Bs(512)F(32,32,32)lr(0.0001)** | 14.80 | **13.69** | Bs(512)L(32,32,32)lr(0.0001) | 50.75 | 48.87 |
| **Bs(512)F(64,64,64)lr(0.0001)** | **14.72** | 14.40 | Bs(512)L(64,64,64)lr(0.0001) | 23.92 | 17.66 |
| Bs(512)F(64,64,64)lr(0.01) | 16.10 | 16.75 | **Bs(512)L(32,32,32)lr(0.01)** | **13.82** | **13.67** |
| Bs(512)F(32,32,32)lr(0.01) | 16.43 | 16.88 | Bs(512)L(64,64,64)lr(0.01) | 41.85 | 41.62 |
| Bs(128)F(64,64,64)lr(0.0001) | 14.78 | 13.73 | Bs(128)L(32,32,32)lr(0.0001) | 15.88 | 15.11 |
| Bs(128)F(64,64,64)lr(0.01) | 17.52 | 19.17 | Bs(128)L(32,32,32)lr(0.01) | 21.15 | 15.15 |

Table 4.4: Hyperparameters selection for FD002 and FD004

| CNN hyperparameters | FD002 RMSE | FD004 RMSE | LSTM hyperparameters | FD002 RMSE | FD004 RMSE |
|---|---|---|---|---|---|
| Bs(512)F(16,16,16)lr(0.001) | 23.03 | 24.86 | Bs(512)F(16,16,16)lr(0.001) | 18.04 | 18.86 |
| Bs(512)F(16,16,16)lr(0.0001) | 18.40 | 23.51 | Bs(512)F(16,16,16)lr(0.0001) | 41.52 | 40.39 |
| Bs(128)F(16,16,16)lr(0.001) | 23.64 | 25.11 | **Bs(128)F(16,16,16)lr(0.001)** | **17.81** | **18.68** |
| **Bs(128)F(16,16,16)lr(0.0001)** | **18.27** | **22.91** | Bs(128)F(16,16,16)lr(0.0001) | 17.85 | 19.05 |
| Bs(128)F(32,32,32)lr(0.0001) | 21.18 | 24.44 | Bs(128)F(32,32,32)lr(0.001) | 19.14 | 19.85 |
| Bs(128)F(64,64,64)lr(0.0001) | 21.13 | 24.37 | Bs(128)F(64,64,64)lr(0.001) | 19.47 | 22.28 |

Since neural networks learning is non convex, the models with the selected hyperparameters are trained 10 times, and the best models according to the validation are selected, in this step, the training set is divided randomly 90% for training and 10%

for validation. The RUL prediction accuracies for the test units by using the CNN model, the LSTM model, and the deep ensemble method are presented in Table 4.5.

Table 4.5: Evaluation of the CNN, LSTM, and ensemble model

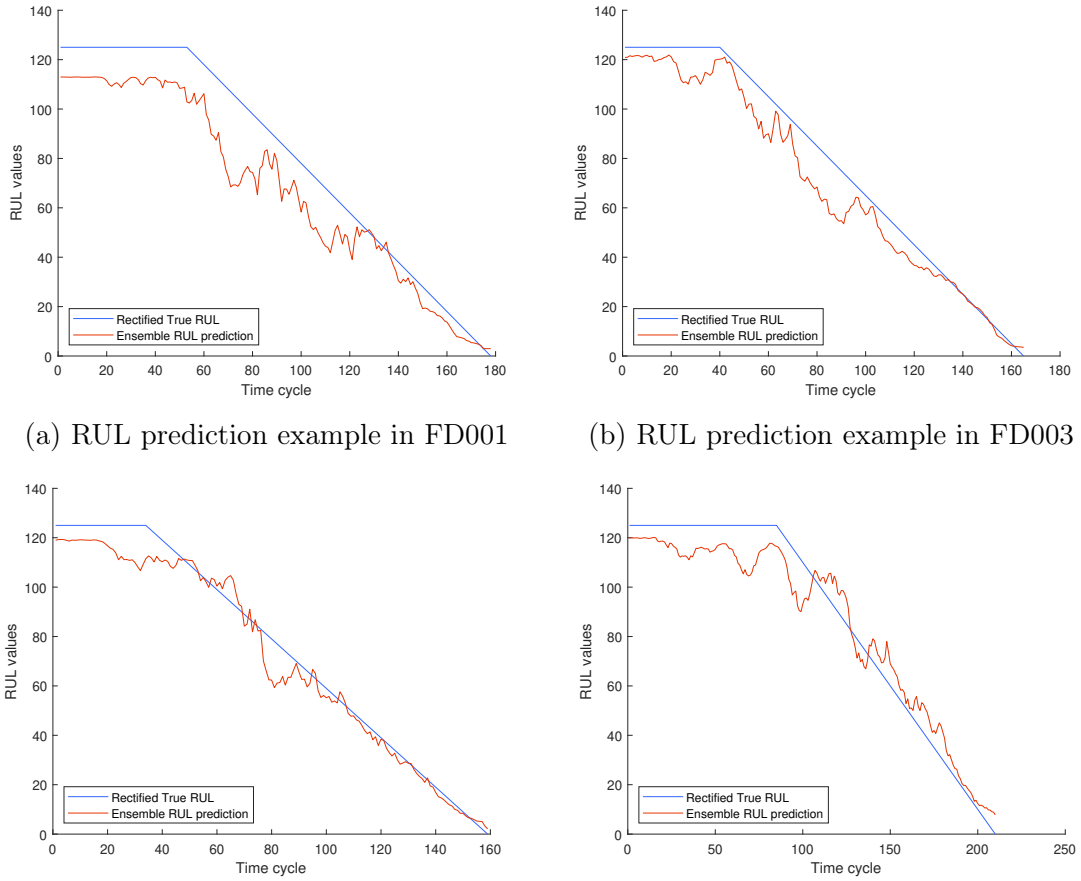
| Model | FD001 | | FD003 | | FD002 | | FD004 | |
|---|---|---|---|---|---|---|---|---|
| | RMSE | Score | RMSE | Score | RMSE | Score | RMSE | Score |
| CNN | 13.42 | 227.20 | 13.43 | 266.28 | 27.64 | 11269.07 | 33.01 | 12058.89 |
| LSTM | 13.53 | 286.16 | 14.24 | 284.59 | 28.05 | 12322.89 | 30.02 | 9246.61 |
| **Ensemble CNN-LSTM** | **12.61** | **218.68** | **13.17** | **244.28** | **27.44** | **11061.08** | **30.18** | **7418.26** |

The separated CNN and LSTM show a good RUL accuracy (RMSE and Score), while the ensemble of CNN and LSTM reveals better accuracy than each separated model. Figure 4-12 shows examples of predicted RUL for each subdataset, the sequences are taken from the validation data. The x-axis is the running time while the y-axis is the RUL values. The blue line represents the rectified true RUL while the red line represents the predicted RUL, it can be seen that the RUL prediction is more accurate with time and becomes more precise near the failure. The figure also shows that the RUL is well predicted even when there are different operating condition modes in FD002 and FD004.

The RUL prediction results of the testing engine units for each subdataset are presented in Figure 4-13. The true RULs of the testing units are sorted from small to large in order to enhance the results visualization.

The x-axis represents the testing units, while the y-axis represents the RUL value for each unit. It can be observed that the RUL prediction is more accurate when the engine is near to failure (when the true RUL is small), this is because the degradation is in a significant level and the information about the degradation can be seen in the collected data. Hence, our method is able to capture this degradation level and predict the RUL with high accuracy. Figure 4-13c and 4-13d present a suitable RUL prediction accuracy even when there is a high variability of the operating condition modes (six operating condition modes). Therefore, the proposed approach has proven its efficiency to capture the variability of the different operating condition modes.

The efficiency of the ensemble deep method is shown in Figure 4-14. Figure 4-14b is a zoom of Figure 4-14a for the test unit #60, it shows that the predicted RUL




(a) RUL prediction example in FD001

(b) RUL prediction example in FD003

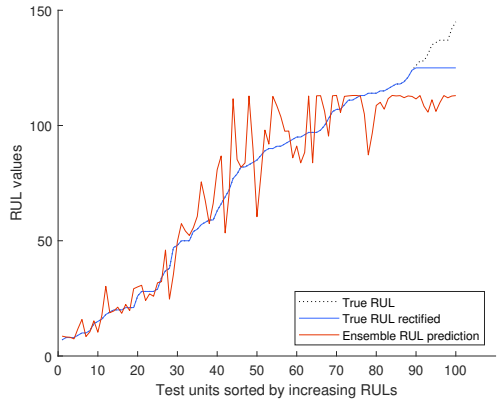
(c) RUL prediction example in FD002

(d) RUL prediction example in FD004

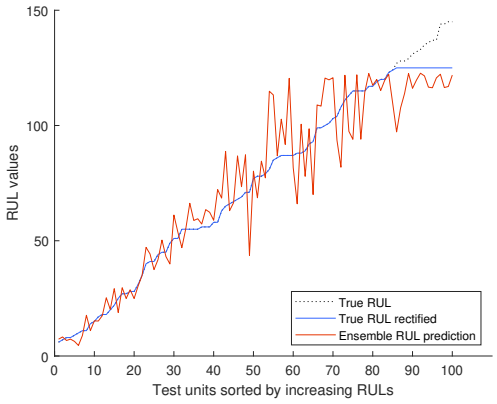
Figure 4-12: RUL prediction examples for the validation sequences using the proposed approach

using each separated model CNN and LSTM is far from the true RUL. However, the proposed deep ensemble method shows a better RUL prediction closer to the true RUL, this is performed thanks to the proposed fusion method where the predicted RULs using each model are aggregated by using the weighted mean.

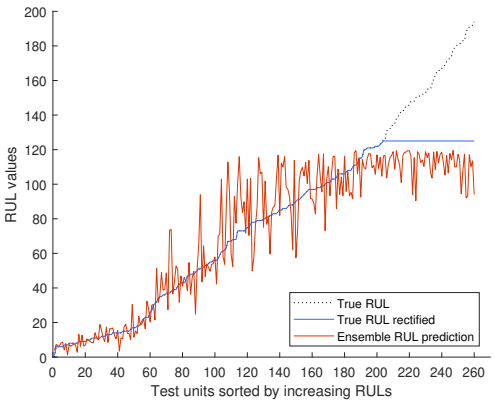
The performance of our proposed approach is compared with the state-of-the-art prognostics approaches applied to the C-MAPSS dataset. Table 4.6 summarizes the latest research results sorted in the ascending order of publication year. The comparison is made in the same conditions, the rectified RUL is applied for the training set, and not in the testing set ($N/A$ refers to the not available information). The rectification is not applied for the test units means that the predicted RUL is compared with the dashed line in Figure 4-13. From Table 4.6, it can be seen that



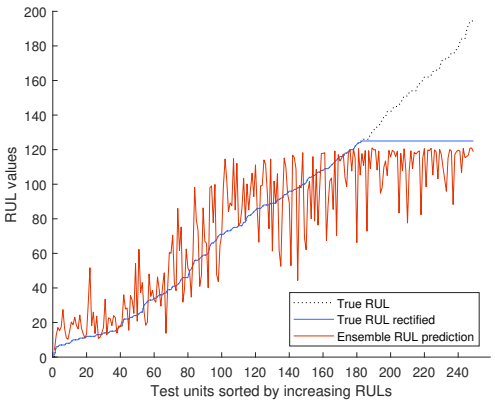

(a) sorted RUL of test units in FD001




(b) sorted RUL of test units in FD003




(c) sorted RUL of test units in FD002

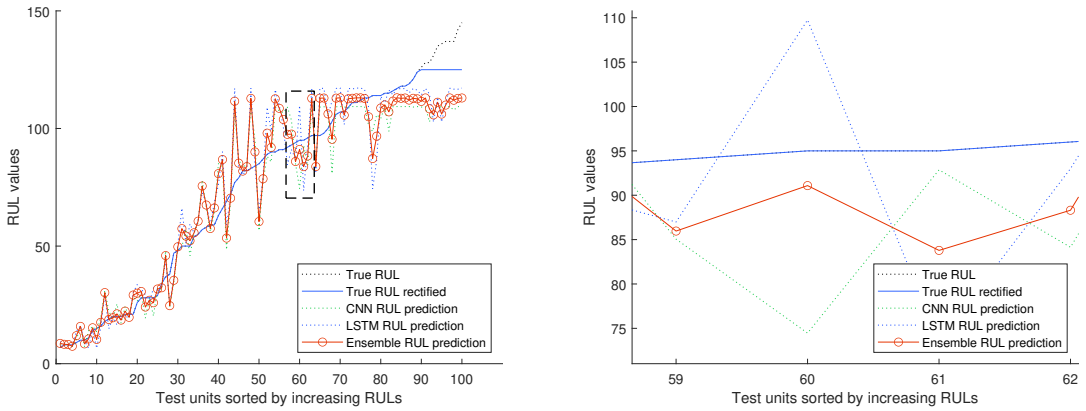



(d) sorted RUL of test units in FD004

Figure 4-13: RUL prediction for the test units for all the subdatasets

recently deep learning models are widely applied and give better results compared to the traditional machine learning models (SVM and MLP). It can also be observed that the RMSE and scoring errors are lower for FD001 and FD003 than FD002 and FD004, this is due to the viability of the operating condition modes in the latter two subdatasets. Our proposed approach outperforms the other related works in terms of accuracy (RMSE and Score are low) for the subdatasets FD001 and FD003, this can confirm the suitability of exploiting the advantage of each model (CNN and LSTM) by fusing their decision (RUL estimation).

However the other works have a slightly better performance for FD002 and FD004, this is because we applied a rectification on the training set (RUL constant = 125), but this rectification is not applied for the test set. There is a slight difference be-




(a) Predicted RUL of test units in FD001 (b) Predicted RUL for test unit #60

Figure 4-14: Predicted RUL for the test unit #60 in FD001

Table 4.6: Performance comparison with the related works on C-MAPSS dataset (without applying rectification on the test set)

| Method | Year | RUL constant on train set | RUL constant on test set | FD001 | | FD003 | | FD002 | | FD004 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | RMSE | Score | RMSE | Score | RMSE | Score | RMSE | Score |
| SVM (Louen et al. 2013) | 2013 | Not applied | Not applied | 29.82 | - | - | - | - | - | - | - |
| MLP (Lim et al. 2016) | 2016 | N/A | Not applied | 15.15 | - | - | - | - | - | - | - |
| CNN (Babu et al. 2016) | 2016 | 130 | Not applied | 18.44 | 1286.70 | 19.81 | 1596.20 | 30.29 | 13570 | 29.15 | 7886.4 |
| LSTM (S. Zheng et al. 2017) | 2017 | 130 | Not applied | 16.14 | 338 | 16.18 | 852 | **24.49** | **4450** | 28.17 | **5550** |
| LSTM (Hsu et al. 2018) | 2018 | N/A | Not applied | 16.73 | 388.68 | 18.06 | 822.19 | 29.43 | 10654 | 28.39 | 6370 |
| BLSTM (J. Zhang et al. 2018) | 2018 | 130 | Not applied | 15.42 | - | - | - | - | - | - | - |
| ELM (C. Zheng et al. 2018) | 2018 | 125 | Not applied | 13.78 | 267.31 | - | - | - | - | - | - |
| CNN (X. Li et al. 2018) | 2018 | 125 | Not applied | 13.32 | - | 14.02 | - | 24.86 | - | 29.44 | - |
| Stacking ensemble (Singh et al. 2019) | 2019 | N/A | Not applied | 16.67 | - | 18.44 | - | 25.57 | - | **26.76** | - |
| Hybrid CNN-BLSTM (Xia et al. 2020) | 2020 | 130 | Not applied | 12.66 | 304.29 | - | - | - | - | - | - |
| **Ensemble CNN-LSTM (proposed approach)** | **2020** | **125** | Not applied | **12.61** | **218.68** | **13.17** | **244.28** | 27.44 | 11061.08 | 30.18 | 7418.26 |

tween the true and the rectified true RUL in the test set for the subdatasets FD001 and FD003, this is why our approach outperforms the other works. There is a significant difference between the true and the rectified true RUL in the test set for the subdatasets FD002 and FD004. Hence, this rectification of RUL constant = 125 applied in the literature for C-MAPSS can provide good results for the subdatasets FD001 and FD003, but this constant RUL should be changed for the case of FD002 and FD004. It is worth mentioning that this rectification is applied only to compare the results of our proposed approach with the related works (to be in the same conditions).

Some works applied the RUL rectification to both the training and the test sets. Table 4.7 compares the obtained results by applying the same conditions. The difference between the target RUL with and without rectification is shown in Figure

4-13.

Table 4.7: Performance comparison with the related works on C-MAPSS dataset (with applying rectification on the test set)

| Method | Year | RUL constant on train set | RUL constant on test set | FD001 | | FD003 | | FD002 | | FD004 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | RMSE | Score | RMSE | Score | RMSE | Score | RMSE | Score |
| BLSTM (J. Wang et al. 2018) | 2018 | 125 | 125 | 13.65 | 295 | 13.74 | 317 | 23.18 | 4130 | 24.86 | 5430 |
| CNN (X. Li et al. 2018) | 2018 | 125 | 125 | 12.61 | 273.7 | 12.64 | 284.1 | 22.36 | 10412 | 23.31 | 12466 |
| BHLSTM (Elsheikh et al. 2019) | 2019 | 130 | 130 | - | 376.64 | - | 1422 | - | - | - | - |
| Hybrid CNN-LSTM (Al-Dulaimi et al. 2019) | 2019 | 125 | 125 | 13.017 | 245 | 12.22 | 287.72 | 15.24 | 1282.42 | 18.156 | 1527.42 |
| MS-DCNN (Han Li et al. 2020) | 2020 | 125 | 125 | 11.44 | 196.22 | 11.67 | 241.89 | 19.35 | 3747 | 22.22 | 4844 |
| **Ensemble CNN-LSTM (proposed approach)** | **2020** | **125** | **125** | **10.74** | **176.36** | **11.48** | **206.53** | **14.23** | **984.34** | **18.05** | **1478.70** |

From Table 4.7, it can be observed that the proposed deep ensemble approach outperforms the other methods. In MS-DCNN (Han Li et al. 2020), a multi-scale deep CNN is applied which uses an ensemble of CNN models with different time window lengths. Our proposed approach has shown better performance than MS-DCNN and other methods, due to combining models with different properties. This is due to the fact that CNN can automatically extract relevant features, and LSTM memorizes long term dependency between the data points. The proposed approach is more efficient for RUL estimation than the hybrid CNN-LSTM (Al-Dulaimi et al. 2019), because it shows better accuracy due to the decision fusion. Comparing to the hybrid CNN-LSTM that predicts only one RUL, our proposed approach is more reliable because two RULs are predicted with two different models. Finally, the predicted RULs are merged in order to obtain the final RUL.

## 4.4 Experimentation using filter clogging dataset

### 4.4.1 Dataset presentation

In industry, particles filtration is a common process to achieve a desired level of purification, the particles in the liquids may lead to performance drop and rapid wear propagation of the mechanical systems. Filtration is an operation which separates the suspended particles from the fluid by using a filtration unit named filter. The filter clogging dataset (PHME 2020) is proposed for data challenge of the fifth European conference of the Prognostics and Health Management Society. The experimental

system for the filter clogging evolution is illustrated in Figure 4-15. It consists of a filter, pump, liquid tanks, tank stirrer, pulsation dampener, pressure and flow rate sensors, and a data acquisition system connected to a computer. The system is a liquid circuit, where the pump flows the liquid from a tank to another through a filter. The circuit includes a dampener in order to eliminate possible pulsations in the flow. Three sensors are installed for instrumentation, two pressure transducers sensors to measure the pressure before and after the filter, and a flow meter sensor for flow rate measurement.

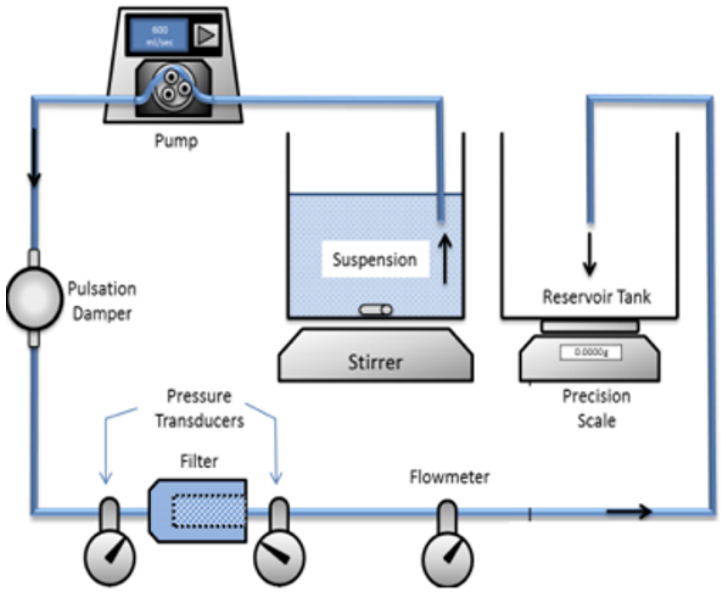



Figure 4-15: Experimental system of filter clogging

The filter has a pore mesh size of 125$\mu m$, as shown in Figure 4-16. The suspension (liquid) is composed of Polyetheretherketone (PEEK) particles and water. The suspension is created by adding particles with different size (small and large), the small particles have a size of 45-53$\mu m$, while the large particles have a size of 63-75$\mu m$.

For each particles size, the experimentation is run with different suspension concentrations as illustrated in Table 4.8. The particles quantity is fixed (32g) while the quantity of water in the suspension tank is changed. Consequently, the concentration of particles (solid ratio) is changed, the solid ratio is varied from 0.4%, 0.425%, 0.45%, and 0.475%. This is applied in order to obtain different filter clogging

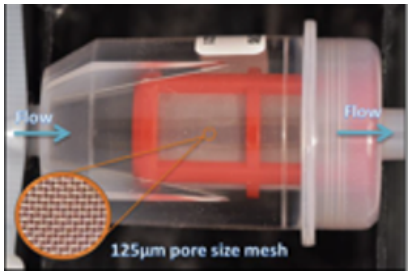



Figure 4-16: The filter used for the dataset

Table 4.8: Suspension details for generating the dataset

| | Concentration number | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| Water (g) in the suspension tank | 7968 | 7497 | 7079 | 6704 |
| Particles (g) | 32 | 32 | 32 | 32 |
| Solid ratio | 0.004<br>0.4% | 0.00425<br>0.425% | 0.0045<br>0.45% | 0.00475<br>0.475% |

evolution (different degradation dynamics).

The experimentation is run 4 times with each concentration number. Hence, for each particles size (small and large), 16 experimentation are achieved (16 run-to-failure sequences are generated). The concentration numbers 1, 2, and 3 are used for training, while the concentration number 4 is used for testing as shown in Table 4.9 and Table 4.10, the training set includes 24 run-to-failure sequences while the testing set contains 8 run-to-failure sequences.

Table 4.9: Training set

| Particles size | Concentration number | Solid ratio | Number of sequences |
|---|---|---|---|
| Small<br>45-53 | 1 | 0.4 | 4 |
| | 2 | 0.425 | 4 |
| | 3 | 0.45 | 4 |
| Large<br>63-75 | 1 | 0.4 | 4 |
| | 2 | 0.425 | 4 |
| | 3 | 0.45 | 4 |

The dataset is composed of the time index, the flow rate of the liquid, the pressure before the filter (upstream pressure), and the pressure after the filter (downstream pressure), the measurements are acquired at a frequency of 10Hz. The failure threshold is set by computing the pressure drop, which is equal to upstream pressure –

Table 4.10: Testing set

| Particles size | Concentration number | Solid ratio | Number of sequences |
|---|---|---|---|
| Small 45-53 | 4 | 0.475 | 4 |
| Large 63-75 | 4 | 0.475 | 4 |

Table 4.11: Filter clogging data format

| Column n# | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Information | Time index | Flow rate | Upstream pressure | Downstream pressure |

downstream pressure. The filter is considered clogged (failure) when the pressure drop is higher than 20 psi (pound per square inch equivalent to 6894.76 Pascal). The data format is presented in Table 4.11.

The objective of this application example is to compute the RUL from the starting of the experiment (sequence). The RUL should be predicted every 1 second (every 10 samples), and the Mean Absolute Error (MAE) is used for the evaluation of the predicted RUL (given as metric by the data challenge), which is computed as follow:

$$MAE = \frac{1}{n_t} \sum_{t=1}^{n_t} |rul(t) - true\_rul(t)| \tag{4.12}$$

where $t$ is the time index, and $n_t$ is the time of failure (end of the sequences). $rul(t)$ is the predicted RUL and $true\_rul$ is the true RUL.

### 4.4.2 Results and discussion

In the offline phase, the collected data from sensors are preprocessed before starting the training. An example of the collected raw data is presented in Figure 4-17. The figure presents the data collected from an experiment with small particles size and concentration number 1 (solid ratio = 0.4%). The collected data is a run-to-failure sequence from the healthy state until the clogging of the filter.

The collected data are sub-sampled by computing the mean for each 10 samples (1 second) in order to obtain 1 sample for each second. This permits to compute a large sliding window with few points. For example, a 10 seconds window contains

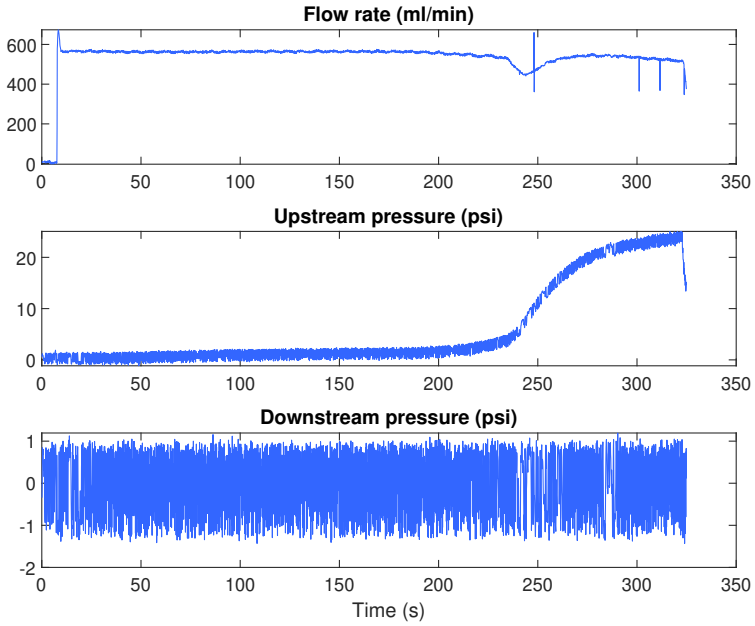



Figure 4-17: Example of collected raw data for filter clogging

100 samples without sub-sampling and 10 samples with sub-sampling. In addition, the computed mean will also smooth the collected data and remove the noises. Then, the drop pressure is computed (pressure drop = upstream pressure - downstream pressure) in order to set a failure threshold on the sequences (when the pressure drop is higher than 20 psi). Figure 4-18 shows the example of collected raw data after subsampling using the mean and after deleting the points that exceed the given failure threshold.

The pressure drop for all the sequences (training and test sequences) is shown in Figure 4-19, where the experiments with small particles size are presented in Figure 4-19a, and the experiments with large particles size are presented in Figure 4-19b. The training sequences are represented with dark color, whereas the testing sequences are represented with the lighter color. From the two figures, it can be observed that the degradation evolution of the test sequences is faster than the train sequences, this fast evolution is due to the concentration of the particles in the test sequences which is higher than the train sequences. Therefore, the filter clogging is faster when more particles are present in the liquid and also when the particles size are large.

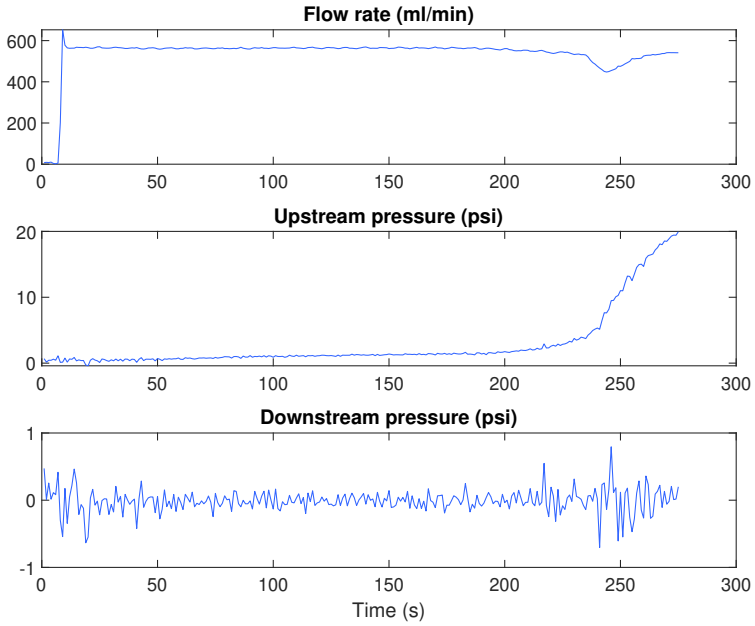
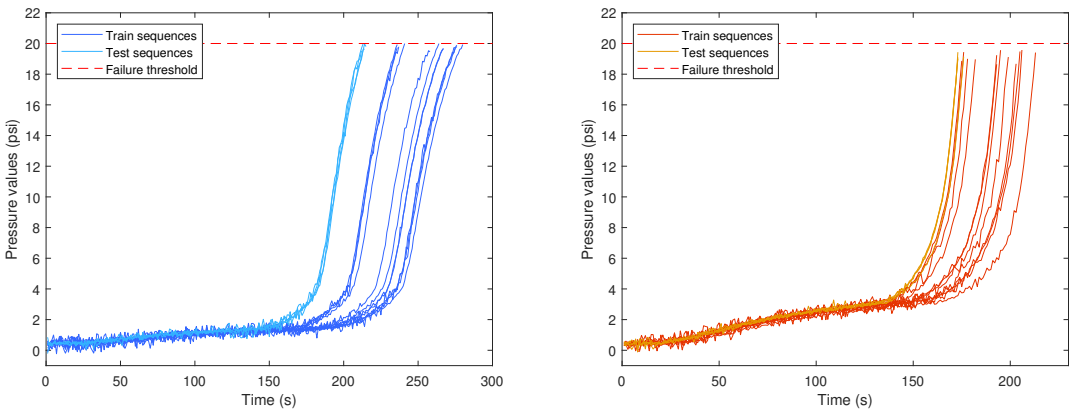



Figure 4-18: Example of preprocessed raw data




(a) Pressure drop for small particles

(b) Pressure drop for large particles

Figure 4-19: Computed pressure drop for small and large particles

For the data challenge, the particles size and the concentration number can be used as input. However, this information about the particles size and concentration number should be known in advance. In the case where this information cannot be provided, the RUL will be predicted using only the sensors data as input (flow rate, upstream pressure, downstream pressure), as well as the pressure drop. Hence, the information about the particles size and the concentration are not provided to the model as input.

The selected inputs are then normalized using the z-score normalization (see equation 4.1), which are next reshaped using a sliding time window, as shown in Figure 4-2. The true RUL (output) is also computed for the training set, which is inversely proportional to the running time.

The goal is to predict the RUL as soon as possible from the start of the experiment (sequence). Hence, the time window length should be small and contains sufficient information for prediction. Figure 4-20 represents a zoom on the pressure drop feature in order to observe the beginning of the degradation. The window length should be small in order to start the RUL prediction as soon as possible and not large in order to avoid prediction delay. As it can be seen in Figure 4-20, the drop pressure feature is constant in the beginning for about 30 seconds, choosing a window smaller than 30 seconds is not useful because the predicted RUL will not be accurate (no variation on the features), and choosing a time window larger than 30 seconds will add a delay for starting the prediction. Hence the optimal time window length for this experiment is 30 seconds and it is chosen for the RUL estimation.

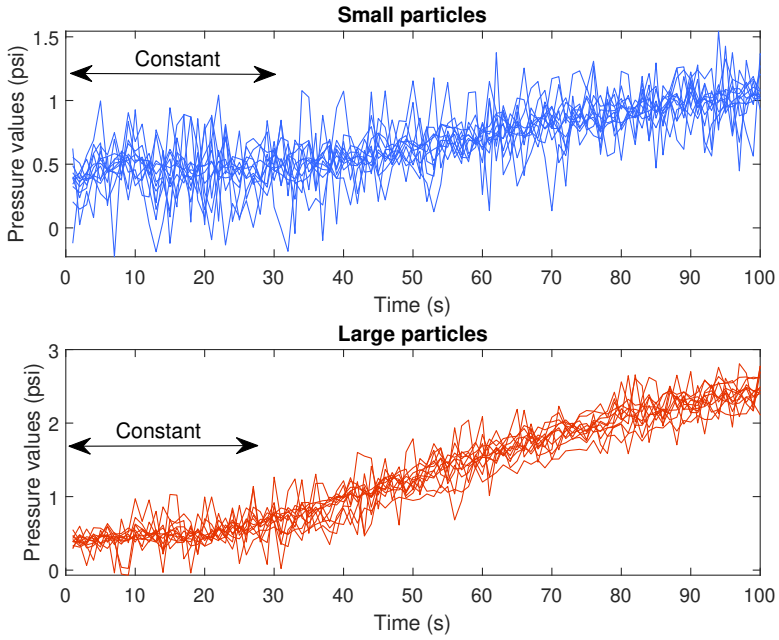



Figure 4-20: Start of the sequences

For training, the mean absolute error is used as cost function (see Eq. (4.12)).

Table 4.12: Cross validation results using different hyperparameters setting

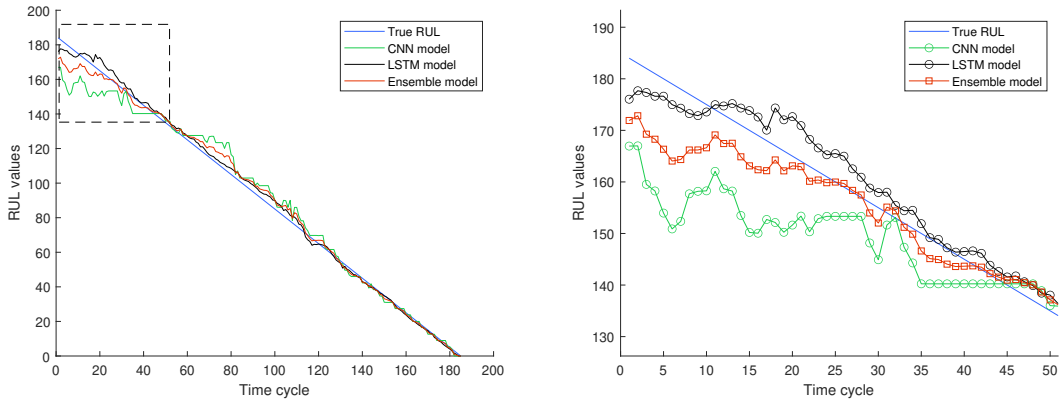
| CNN hyperparameters | MAE | LSTM hyperparameters | MAE |
|---|---|---|---|
| Bs(128)F(32,32,32)lr(0.001) | 14.16 | Bs(128)F(32,32,32)lr(0.001) | 14.61 |
| Bs(128)F(128,128,128)lr(0.001) | 14.55 | Bs(128)F(64,64,64)lr(0.001) | 13.16 |
| **Bs(512)F(32,32,32)lr(0.01)** | **12.78** | Bs(128)F(128,128,128)lr(0.01) | 48.54 |
| Bs(512)F(32,32,32)lr(0.001) | 16.43 | **Bs(128)F(128,128,128)lr(0.001)** | **11.78** |
| Bs(512)F(64,64,64)lr(0.001) | 18.66 | Bs(512)F(128,128,128)lr(0.01) | 46.20 |
| Bs(512)F(128,128,128)lr(0.001) | 17.70 | Bs(512)F(128,128,128)lr(0.001) | 16.34 |

Adam optimizer algorithm (Kingma et al. 2014) is used for training the two models CNN and LSTM. An early stopping criterion is used in order to stop the training when the validation error is not decreasing within 10 iterations. K-fold cross validation is applied in order to select the optimal model hyperparameters. The model is trained using sequences generated with two concentration numbers and validated using the third one (K=3), for example, the model is trained with the sequences from concentration number 1 and 2 (0.4% and 0.425%), then it is validated using the remaining sequences from concentration number 3 (0.45%) using the small and large particles size. This is done in order to find the model that can be generalized for different concentration numbers, knowing that the concentration number in the test sequences is different from the training sequences. Table 4.12 shows the MAE errors of RUL prediction with different combinations of hyperparameters induced by the cross validation technique. The varied hyperparameters are mini-batch size $Bs$, learning rate $lr$, filters number $F$ for CNN, and hidden units number $L$ in LSTM. The best hyperparameters combination according to the RUL prediction error are shown in the table with bold font.

Table 4.13: Prediction errors using CNN, LSTM, and ensemble

| Model | MAE |
|---|---|
| CNN | 6.98 |
| LSTM | 6.32 |
| **Ensemble CNN-LSTM** | **6.19** |

Table 4.13 shows the RUL prediction errors MAE for the testing sequences. It



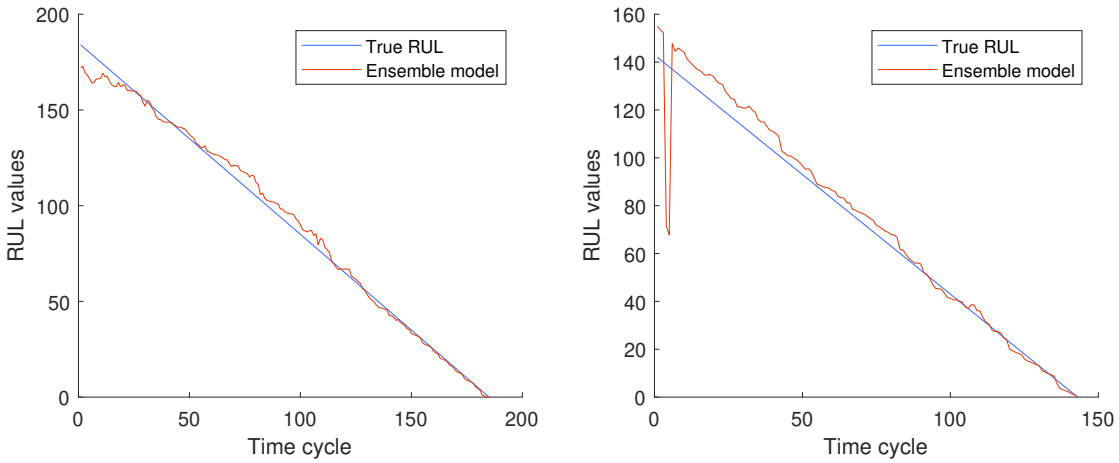

(a) Predicted RUL for filter with small particles size

(b) Zoom on the beginning of the Predicted RUL

Figure 4-21: Predicted RUL for filter with small particles size in the test set

can be seen that the performance of the deep ensemble model is better than the performance of each model (CNN and LSTM). This confirms the efficiency of using an ensemble of models, which can improve the accuracy of the RUL prediction. Figure 4-21a shows the predicted RUL for the test set with small particles size, where the predicted RUL shows a good accuracy comparing to the true RUL. Figure 4-21b represents a zoom on the rectangle with dashed points, this figure shows that the proposed ensemble method improves the performance of the RUL prediction, thanks to the weighted mean fusion of the predicted RULs.

An example of RUL prediction for the filter with small and large particles is presented in Figure 4-22a and 4-22b, respectively. It can be observed that the predicted RUL is less accurate at the beginning of prediction then it becomes more accurate when it is near to the failure, this is because when the degradation is at a low level (in the beginning) there no sufficient information in the features in order to predict the RUL. However, when the degradation is at a high level (near failure), the information about the degradation level can be seen on the features (a high variation on the features amplitude). In Figure 4-22b, there is a high variation of RUL prediction at the beginning of the sequence, this is due to an outlier of prediction caused by the LSTM model.




(a) Predicted RUL for filter with small particles size

(b) Predicted RUL for filter with large particles size

Figure 4-22: Examples of RUL prediction for the filter with small and large particles

## 4.5 Conclusion

In this chapter, a deep ensemble approach is proposed, which exploits the diversity of two different deep learning models named Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). CNN architecture can extract relevant information by applying several convolution filters on the raw data, while LSTM is able to capture the temporal information in time series data. The proposed deep ensemble approach for RUL estimation is validated using the well known C-MAPSS dataset and has achieved promising performance compared with the state-of-the-art results. It is also validated using a filter clogging dataset available through the PHM Europe data challenge. The obtained results on this dataset showed a suitable RUL prediction accuracy which becomes more precise when the degradation is closer to the failure.

Since the reliability of prediction is necessary for the critical industrial systems, the proposed deep ensemble approach can enhance the RUL prediction reliability by fusing the prediction of two different models. The proposed approach has also shown its ability to capture the variability of the different operating condition modes when it is evaluated using the C-MAPSS experimentation (complex dynamic system with

different operation modes). Finally, the proposed deep ensemble method for RUL estimation has proven its efficiency demonstrated through the obtained results since it improved the accuracy of RUL prediction.

# Chapter 5

# General conclusion

## 5.1 Summary of the thesis

System failures can lead to high economic losses for industrial companies due to reparation and downtime costs. Hence, it is necessary to apply an optimal maintenance strategy that can increase the reliability and availability of the industrial systems while reducing the maintenance costs. The maintenance strategy evolved from corrective, preventive, to predictive maintenance strategy (PHM strategy). The predictive maintenance strategy is efficient since it can overcome the limits of the two other strategies by triggering maintenance actions depending on the health conditions of the system. Generally, the pipeline of the PHM strategy includes five steps: data acquisition, data processing, fault diagnostics, fault prognostics, and decision support (health management). Fault prognostics is one of the main steps for achieving PHM strategy, it aims to estimate the Remaining Useful Life (RUL) before failure. It can help to plan the maintenance actions in advance before failure occurrence in order to avoid systems downtime and reduce the revenue losses.

Fault prognostics can be achieved using experience based (reliability and similarity), model based, or data-driven approaches. The latter present the best trade-off in terms of precision, implementation, and cost. Fault prognostics based on data-driven approaches requires sufficient historical degradation sequences covering all degradation dynamic evolution and conditions in order to construct a model that can achieve

a desirable RUL estimation accuracy. Different approaches are proposed in the literature for RUL estimation when multiple historical degradation sequences are available (Heimes 2008; Medjaher et al. 2012; Soualhi et al. 2014; X. Li et al. 2018). In industrial systems, a large quantity of data can be available about the normal operation conditions of the system, while the historical degradation data of the system are often unavailable (e.g., new machines) or insufficient (i.e., few sequences are available). Due to the lack of a priori run-to-failure sequences, several challenges need to be addressed:

- Estimate RUL without any a priori degradation data is challenging. How could online HI selection method be developed to improve this selection over time thanks to the new incoming degradation data? and how to predict its evolution over time until the failure?
- RUL estimation is still difficult when only a few a priori sequences about the degradation that do not cover all degradation dynamic evolution and conditions are available. How could those available degradation sequences be used to improve the accuracy of RUL estimation?
- After collecting several a priori degradation sequences offline, how could the RUL prediction be more accurate? and how to predict the RUL when the system switches between different operating condition modes?

In this manuscript, three data-driven approaches are proposed for achieving fault prognostics. The first two proposed approach aims at predicting the RUL when a priori degradation sequences are insufficient (without and with few a priori sequences). The third approach aims at predicting the RUL when multiple a priori degradation sequences are available. The proposed approaches permit to address the aforementioned challenges by using three RUL estimations paths depending on the quantity of historical degradation data:

- The blind path is triggered when no a priori sequences are available, it aims to select the best HI dynamically using a selection criterion based on the goodness

of fit with the GLM model. At the same time, the Generalized Linear Model (GLM) is used in an adaptive manner in order to extrapolate the HI until failure threshold. The results showed that the RUL prediction is improved over time with the arrival of new incoming degradation data.

- The informed path is triggered when one or more a priori sequences are available. Different RULs are estimated using different adaptive models combined with different HIs, while the final RUL is obtained by merging the RULs using a weighted mean. The informed path showed better RUL prediction results when few a priori sequences are available compared with the methods applied in the literature (SVR and LSTM). Also, the RUL estimation accuracy and robustness are improved when more a priori sequences are collected. Besides, it can predict the RUL when the degradation speed (dynamics) of the incoming sequence is different from the a priori degradation speed, or dynamics thanks to the use of different adaptive models.

- The deep ensemble approach is used when many a priori degradation sequences are available. Two RULs are predicted using two deep learning models named Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM), where the final RUL is obtained by merging the two RUL using weighted mean. The proposed deep ensemble approach has shown promising results against the related works in the state-of-the-art. Also, it has the ability to capture the variability of the different operating condition modes.

## 5.2 Open issues and future directions

- **Switching mechanism.** The informed path is triggered for RUL estimation when few (insufficient) a priori sequences are available offline. The prediction is improved when more a priori sequences are available. The deep ensemble approach or deep path is used for RUL prediction when multiple a priori sequences are available. Hence, one future work is to add a criterion that allows

to switch from informed path to deep path when there is sufficient number of degradation sequences in order to cover all the degradation dynamic evolution and conditions, as illustrated in Figure 5-1. The switching criterion can be based on the RUL prediction accuracy of each path on the available offline sequences. Also, it is intended to investigate the use of both paths (informed and deep) in parallel and merging their predicted RULs in order to obtain a better, precise, and reliable RUL.

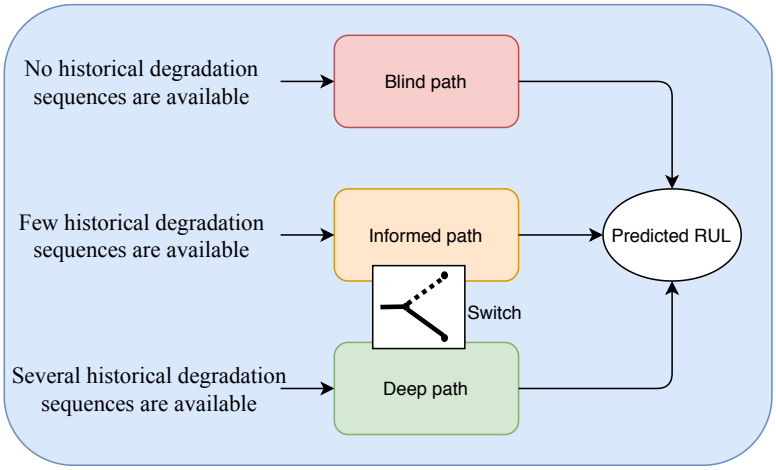



Figure 5-1: Illustration for switching from the informed path to the deep path

- **Failure threshold and confidence interval.** In this manuscript, failure threshold setting and confidence interval estimation were not investigated. In order to predict the RUL when no and few historical degradation sequences are available, it is necessary to set a failure threshold. In the blind path, the solution is to set the threshold according to human expert or feedback from other similar systems since no a priori sequences are available for automatically select the threshold. On the other hand, the threshold can be set in the informed path automatically using the few available a priori sequences. A confidence interval should be given for the predicted RUL, because it is difficult to estimate the RUL with complete precision due to multiple uncertainty sources that influence the prediction. According to (Sankararaman et al. 2013), different sources of uncertainty should be considered: present uncertainty (the estimation of the present system conditions should be precise), future uncertainty (the future

operating conditions are not known since the future is unknown), modeling uncertainty (the HI modeling and the failure threshold are uncertain due to the model parameters estimation and process noise), and prediction method uncertainty (it is necessary to quantify the combined effects of the different uncertainty sources on the predicted RUL).

- **Transfer learning.** It is a future work direction that can deal with insufficient a priori degradation sequences. It aims at transferring the knowledge, in terms of model parameters, weights, etc.) between related problems. In RUL prediction, transfer learning exploits data from other related systems in order to train a model that permits to predict the RUL. Then, the trained model is tuned with the few available a priori sequences of the new system for achieving the RUL prediction. Also, the transfer learning can be used for transferring the information from an operating condition mode to another in the same system (A. Zhang et al. 2018).

# Bibliography


Abid, Koceila, Moamar Sayed-Mouchaweh, and Laurence Cornez. "Adaptive Machine Learning Approach for Fault Prognostics based on Normal Conditions-Application to Shaft Bearings of Wind Turbine". In: *Proceedings of the Annual Conference of the PHM Society*. Vol. 11. 1. 2019.

— "Fault Prognostics for the Predictive Maintenance of Wind Turbines: State of the Art". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2018, pp. 113–125.

Abid, Koceila et al. *Simulated Bearing Degradation Data*. https://figshare.com/articles/Simulated_Bearing_Degradation_Data_mat/12554690/2. June 2020. DOI: 10.6084/m9.figshare.12554690.v1.

Ahmad, Wasim, Sheraz Ali Khan, and Jong-Myon Kim. "A hybrid prognostics technique for rolling element bearings using adaptive predictive models". In: *IEEE Transactions on Industrial Electronics* 65.2 (2017), pp. 1577–1584.

Ali, Jaouher Ben et al. "Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network". In: *Mechanical Systems and Signal Processing* 56 (2015), pp. 150–172.

An, Jinwon and Sungzoon Cho. "Variational autoencoder based anomaly detection using reconstruction probability". In: *Special Lecture on IE* 2.1 (2015).

Antoni, Jérôme. "Cyclic spectral analysis of rolling-element bearing signals: Facts and fictions". In: *Journal of Sound and vibration* 304.3-5 (2007), pp. 497–529.

— "The spectral kurtosis: a useful tool for characterising non-stationary signals". In: *Mechanical systems and signal processing* 20.2 (2006), pp. 282–307.

Atamuradov, Vepa et al. “Railway point machine prognostics based on feature fusion and health state assessment”. In: *IEEE Transactions on Instrumentation and Measurement* 68.8 (2018), pp. 2691–2704.

Aye, SA and PS Heyns. “An integrated Gaussian process regression for prediction of remaining useful life of slow speed bearings based on acoustic emission”. In: *Mechanical Systems and Signal Processing* 84 (2017), pp. 485–498.

Babu, Giduthuri Sateesh, Peilin Zhao, and Xiao-Li Li. “Deep convolutional neural network based regression approach for estimation of remaining useful life”. In: *International conference on database systems for advanced applications*. Springer. 2016, pp. 214–228.

Bangalore, Pramod and Lina Bertling Tjernberg. “An artificial neural network approach for early fault detection of gearbox bearings”. In: *IEEE Transactions on Smart Grid* 6.2 (2015), pp. 980–987.

Bechhoefer, Eric, Brandon Van Hecke, and David He. “Processing for improved spectral analysis”. In: *Annual Conference of the Prognostics and Health Management Society, New Orleans, LA, Oct.* 2013, pp. 14–17.

Bengio, Yoshua, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

Benkedjouh, Tarak et al. “Remaining useful life estimation based on nonlinear feature reduction and support vector regression”. In: *Engineering Applications of Artificial Intelligence* 26.7 (2013), pp. 1751–1760.

Boškoski, Pavle et al. “Bearing fault prognostics using Rényi entropy based features and Gaussian process models”. In: *Mechanical Systems and Signal Processing* 52 (2015), pp. 327–337.

Callan, Rob, Brian Larder, and John Sandiford. “An integrated approach to the development of an intelligent prognostic health management system”. In: *2006 IEEE Aerospace Conference*. IEEE. 2006, 12–pp.

Chammas, Antoine et al. “Drift detection and characterization for condition monitoring: application to dynamical systems with unknown failure modes”. In: *IMA Journal of Management Mathematics* 26.2 (2015), pp. 225–243.

Chandola, Varun, Arindam Banerjee, and Vipin Kumar. “Anomaly detection: A survey”. In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.

Cheng, Fangzhou, Liyan Qu, and Wei Qiao. “Fault prognosis and remaining useful life prediction of wind turbine gearboxes using current signal analysis”. In: *IEEE Transactions on Sustainable Energy* 9.1 (2017), pp. 157–167.

Coble, Jamie Baalis. “Merging data sources to predict remaining useful life–an automated method to identify prognostic parameters”. In: (2010).

— “Merging data sources to predict remaining useful life–an automated method to identify prognostic parameters”. In: (2010).

Coble, Jamie and J Wesley Hines. “Identifying optimal prognostic parameters from data: a genetic algorithms approach”. In: *Annual conference of the prognostics and health management society*. Vol. 27. 2009.

Das, Sreerupa et al. “An open architecture for enabling CBM/PHM capabilities in ground vehicles”. In: *2012 IEEE Conference on Prognostics and Health Management*. IEEE. 2012, pp. 1–8.

Duda, Richard O, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

Al-Dulaimi, Ali et al. “A multimodal and hybrid deep neural network model for remaining useful life estimation”. In: *Computers in Industry* 108 (2019), pp. 186–196.

Duong, Bach Phi et al. “A Reliable Health Indicator for Fault Prognosis of Bearings”. In: *Sensors* 18.11 (2018), p. 3740.

Dupuis, Richard. “Application of oil debris monitoring for wind turbine gearbox prognostics and health management”. In: *Annual Conference of the prognostics and health management society*. 2010, pp. 10–16.

Eker, Ömer Faruk, Faith Camci, and Ian K Jennions. “Major challenges in prognostics: study on benchmarking prognostic datasets”. In: (2012).

Elsheikh, Ahmed, Soumaya Yacout, and Mohamed-Salah Ouali. "Bidirectional handshaking LSTM for remaining useful life prediction". In: *Neurocomputing* 323 (2019), pp. 148–156.

Fernandez-Francos, Diego et al. "Automatic bearing fault diagnosis based on one-class $\nu$-SVM". In: *Computers & Industrial Engineering* 64.1 (2013), pp. 357–365.

Gebraeel, Nagi. "Sensory-updated residual life distributions for components with exponential degradation patterns". In: *IEEE Transactions on Automation Science and Engineering* 3.4 (2006), pp. 382–393.

Gebraeel, Nagi et al. "Residual life predictions from vibration-based degradation signals: a neural network approach". In: *IEEE Transactions on industrial electronics* 51.3 (2004), pp. 694–700.

Goebel, K et al. "Prognostic Performance Metrics". In: *Machine Learning and Knowledge Discovery for Engineering Systems Health Management*. Ed. by A. N. Srivastava and J. Han. Chapman and Hall - CRC, 2011. Chap. 5.

Goh, KM et al. "A review of research in manufacturing prognostics". In: *2006 4th IEEE International Conference on Industrial Informatics*. IEEE. 2006, pp. 417–422.

Guo, Liang et al. "A recurrent neural network based health indicator for remaining useful life prediction of bearings". In: *Neurocomputing* 240 (2017), pp. 98–109.

Guo, Liang et al. "Machinery health indicator construction based on convolutional neural networks considering trend burr". In: *Neurocomputing* 292 (2018), pp. 142–150.

Guo, Xiaojie, Liang Chen, and Changqing Shen. "Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis". In: *Measurement* 93 (2016), pp. 490–502.

Haidong, Shao et al. "Intelligent fault diagnosis of rolling bearing using deep wavelet auto-encoder with extreme learning machine". In: *Knowledge-Based Systems* 140 (2018), pp. 1–14.

He, Q Peter and Jin Wang. "Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes". In: *IEEE transactions on semiconductor manufacturing* 20.4 (2007), pp. 345–354.

Heimes, Felix O. "Recurrent neural networks for remaining useful life estimation". In: *2008 international conference on prognostics and health management*. IEEE. 2008, pp. 1–6.

Helle, Aino. "Development of prognostic concepts and tools". In: *VTT symposium*. Vol. 243. VTT; 1999. 2006, p. 5.

Heng, Aiwina et al. "Intelligent condition-based prediction of machinery reliability". In: *Mechanical Systems and Signal Processing* 23.5 (2009), pp. 1600–1614.

Hinton, Geoffrey E and Richard S Zemel. "Autoencoders, minimum description length and Helmholtz free energy". In: *Advances in neural information processing systems*. 1994, pp. 3–10.

Hochreiter, Sepp and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

Hsu, Che-Sheng and Jehn-Ruey Jiang. "Remaining useful life estimation using long short-term memory deep learning". In: *2018 IEEE International Conference on Applied System Invention (ICASI)*. IEEE. 2018, pp. 58–61.

Huang, Guang-Bin, Qin-Yu Zhu, and Chee-Kheong Siew. "Extreme learning machine: theory and applications". In: *Neurocomputing* 70.1-3 (2006), pp. 489–501.

Janssens, Olivier et al. "Convolutional neural network based fault detection for rotating machinery". In: *Journal of Sound and Vibration* 377 (2016), pp. 331–345.

Jardine, Andrew KS, Daming Lin, and Dragan Banjevic. "A review on machinery diagnostics and prognostics implementing condition-based maintenance". In: *Mechanical systems and signal processing* 20.7 (2006), pp. 1483–1510.

Javed, Kamran et al. "Enabling health monitoring approach based on vibration data for accurate prognostics". In: *IEEE Transactions on Industrial Electronics* 62.1 (2014), pp. 647–656.

Jiang, Guoqian et al. “Wind turbine fault detection using a denoising autoencoder with temporal information”. In: *IEEE/Asme transactions on mechatronics* 23.1 (2017), pp. 89–100.

Jin, Xiaohang et al. “Anomaly detection and fault prognosis for bearings”. In: *IEEE Transactions on Instrumentation and Measurement* 65.9 (2016), pp. 2046–2054.

Kacprzynski, GJ et al. “Predicting remaining life by fusing the physics of failure modeling with diagnostics”. In: *JOm* 56.3 (2004), pp. 29–35.

Khan, Shehroz S and Michael G Madden. “One-class classification: taxonomy of study and review of techniques”. In: *The Knowledge Engineering Review* 29.3 (2014), pp. 345–374.

Kim, Nam-Ho, Dawn An, and Joo-Ho Choi. *Prognostics and health management of engineering systems: An introduction*. springer, 2016.

Kingma, Diederik P and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

Laouti, Nassim, Nida Sheibat-Othman, and Sami Othman. “Support vector machines for fault detection in wind turbines”. In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 7067–7072.

Le Son, Khanh et al. “Remaining useful life estimation based on stochastic deterioration models: A comparative study”. In: *Reliability Engineering & System Safety* 112 (2013), pp. 165–175.

LeCun, Yann, Yoshua Bengio, et al. “Convolutional networks for images, speech, and time series”. In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

LeCun, Yann et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

Lee, Jay et al. “Intelligent prognostics tools and e-maintenance”. In: *Computers in industry* 57.6 (2006), pp. 476–489.

Lee, Jay et al. “Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications”. In: *Mechanical systems and signal processing* 42.1-2 (2014), pp. 314–334.

Lei, Jinhao, Chao Liu, and Dongxiang Jiang. "Fault diagnosis of wind turbine based on Long Short-term memory networks". In: *Renewable energy* 133 (2019), pp. 422–432.

Lei, Yaguo, Naipeng Li, and Jing Lin. "A new method based on stochastic process models for machine remaining useful life prediction". In: *IEEE Transactions on Instrumentation and Measurement* 65.12 (2016), pp. 2671–2684.

Lei, Yaguo et al. "A model-based method for remaining useful life prediction of machinery". In: *IEEE Transactions on Reliability* 65.3 (2016), pp. 1314–1326.

Lei, Yaguo et al. "Machinery health prognostics: A systematic review from data acquisition to RUL prediction". In: *Mechanical Systems and Signal Processing* 104 (2018), pp. 799–834.

Li, Han et al. "Remaining useful life prediction using multi-scale deep convolutional neural network". In: *Applied Soft Computing* 89 (2020), p. 106113.

Li, Hui, Yucai Wang, and Yanfang Ma. "Ensemble empirical mode decomposition and Hilbert-Huang transform applied to bearing fault diagnosis". In: *2010 3rd International Congress on Image and Signal Processing*. Vol. 7. IEEE. 2010, pp. 3413–3417.

Li, Xiang, Qian Ding, and Jian-Qiao Sun. "Remaining useful life estimation in prognostics using deep convolution neural networks". In: *Reliability Engineering & System Safety* 172 (2018), pp. 1–11.

Li, Xiang, Wei Zhang, and Qian Ding. "Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction". In: *Reliability Engineering & System Safety* 182 (2019), pp. 208–218.

Li, YSCTS et al. "Adaptive prognostics for rolling element bearing condition". In: *Mechanical systems and signal processing* 13.1 (1999), pp. 103–113.

Liao, Linxia. "Discovering prognostic features using genetic programming in remaining useful life prediction". In: *IEEE Transactions on Industrial Electronics* 61.5 (2013), pp. 2464–2472.

Lim, Pin, Chi Keong Goh, and Kay Chen Tan. "A time window neural network based framework for remaining useful life estimation". In: *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2016, pp. 1746–1753.

Louen, C, SX Ding, and C Kandler. "A new framework for remaining useful life estimation using support vector machine classifier". In: *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*. IEEE. 2013, pp. 228–233.

Loutas, Theodoros H, Dimitrios Roulias, and George Georgoulas. "Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic e-support vectors regression". In: *IEEE Transactions on Reliability* 62.4 (2013), pp. 821–832.

Mahamad, Abd Kadir, Sharifah Saon, and Takashi Hiyama. "Predicting remaining useful life of rotating machinery based artificial neural network". In: *Computers & Mathematics with Applications* 60.4 (2010), pp. 1078–1087.

Malhi, Arnaz, Ruqiang Yan, and Robert X Gao. "Prognosis of defect propagation based on recurrent neural networks". In: *IEEE Transactions on Instrumentation and Measurement* 60.3 (2011), pp. 703–711.

Mao, Wentao et al. "Predicting remaining useful life of rolling bearings based on deep feature representation and long short-term memory neural network". In: *Advances in Mechanical Engineering* 10.12 (2018), p. 1687814018817184.

Marsland, Stephen. "Novelty detection in learning systems". In: *Neural computing surveys* 3.2 (2003), pp. 157–195.

McCullagh, Peter. *Generalized linear models*. Routledge, 2018.

McFadden, PD and JD Smith. "Model for the vibration produced by a single point defect in a rolling element bearing". In: *Journal of sound and vibration* 96.1 (1984), pp. 69–82.

Medjaher, Kamal, Diego Alejandro Tobon-Mejia, and Noureddine Zerhouni. "Remaining useful life estimation of critical components with application to bearings". In: *IEEE Transactions on Reliability* 61.2 (2012), pp. 292–302.

Mosallam, Ahmed, Kamal Medjaher, and Noureddine Zerhouni. "Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction". In: *Journal of Intelligent Manufacturing* 27.5 (2016), pp. 1037–1048.

Niu, Gang. *Data-driven technology for engineering systems health management*. Springer, 2017.

Niu, Gang and Bo-Suk Yang. "Intelligent condition monitoring and prognostics system based on data-fusion strategy". In: *Expert Systems with Applications* 37.12 (2010), pp. 8831–8840.

Odgaard, Peter Fogh, Jakob Stoustrup, and Michel Kinnaert. "Fault tolerant control of wind turbines-a benchmark model". In: *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*. Vol. 1. 1. IFAC Barcelona, Spain. 2009, pp. 155–160.

Oppenheimer, Charles H and Kenneth A Loparo. "Physically based diagnosis and prognosis of cracked rotor shafts". In: *Component and Systems Diagnostics, Prognostics, and Health Management II*. Vol. 4733. International Society for Optics and Photonics. 2002, pp. 122–132.

Paris, Pe and Fazil Erdogan. "A critical analysis of crack propagation laws". In: (1963).

PHME. *PHME DATA CHALLENGE 2020*. Ed. by European Conference of the Prognostics and Health Management Society. Accessed: 07.07.2020. 2020. URL: http://phmeurope.org/2020/data-challenge-2020.

Qian, Yuning, Ruqiang Yan, and Shijie Hu. "Bearing degradation evaluation using recurrence quantification analysis and Kalman filter". In: *IEEE Transactions on Instrumentation and Measurement* 63.11 (2014), pp. 2599–2610.

Rai, VK and AR Mohanty. "Bearing fault diagnosis using FFT of intrinsic mode functions in Hilbert–Huang transform". In: *Mechanical Systems and Signal Processing* 21.6 (2007), pp. 2607–2615.

Ramasso, Emmanuel, Michele Rombaut, and Noureddine Zerhouni. "Joint prediction of continuous and discrete states in time-series based on belief functions". In: *IEEE transactions on cybernetics* 43.1 (2012), pp. 37–50.

Saha, B and K Goebel. "Battery data set". In: *NASA AMES prognostics data repository* (2007).

Saidi, Lotfi, Jaouher Ben Ali, and Farhat Fnaiech. "Application of higher order spectral features and support vector machines for bearing faults classification". In: *ISA transactions* 54 (2015), pp. 193–206.

Saidi, Lotfi et al. "Wind turbine high-speed shaft bearings health prognosis through a spectral Kurtosis-derived indices and SVR". In: *Applied Acoustics* 120 (2017), pp. 1–8.

Sankararaman, Shankar and Kai Goebel. "Why is the remaining useful life prediction uncertain". In: *Annual conference of the prognostics and health management society*. Vol. 2013. 2013.

Saxena, Abhinav and Kai Goebel. "C-MAPSS data set". In: *NASA Ames Prognostics Data Repository* (2008).

Saxena, Abhinav and Ashraf Saad. "Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems". In: *Applied Soft Computing* 7.1 (2007), pp. 441–454.

Saxena, Abhinav et al. "Damage propagation modeling for aircraft engine run-to-failure simulation". In: *2008 international conference on prognostics and health management*. IEEE. 2008, pp. 1–9.

Saxena, Abhinav et al. "Metrics for evaluating performance of prognostic techniques". In: *2008 International Conference on Prognostics and Health Management*. IEEE. 2008, pp. 1–17.

Schölkopf, Bernhard et al. "Support vector method for novelty detection". In: *Advances in neural information processing systems*. 2000, pp. 582–588.

Schömig, Alexander K and Oliver Rose. "On the suitability of the Weibull distribution for the approximation of machine failures". In: *IIE Annual Conference. Proceedings*. Institute of Industrial and Systems Engineers (IISE). 2003, p. 1.

Shao, Yimin and Kikuo Nezu. "Prognosis of remaining bearing life using neural networks". In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 214.3 (2000), pp. 217–230.

Shin, Hyun Joon, Dong-Hwan Eom, and Sung-Shick Kim. "One-class support vector machines—an application in machine fault detection and classification". In: *Computers & Industrial Engineering* 48.2 (2005), pp. 395–408.

Singh, Sandip Kumar, Sandeep Kumar, and JP Dwivedi. "A novel soft computing method for engine RUL prediction". In: *Multimedia Tools and Applications* 78.4 (2019), pp. 4065–4087.

Soualhi, Abdenour, Kamal Medjaher, and Noureddine Zerhouni. "Bearing health monitoring based on Hilbert–Huang transform, support vector machine, and regression". In: *IEEE Transactions on Instrumentation and Measurement* 64.1 (2014), pp. 52–62.

Srivastava, Nitish et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

Sun, Wenjun et al. "A sparse auto-encoder-based deep neural network approach for induction motor faults classification". In: *Measurement* 89 (2016), pp. 171–178.

Tenenbaum, Joshua B, Vin De Silva, and John C Langford. "A global geometric framework for nonlinear dimensionality reduction". In: *science* 290.5500 (2000), pp. 2319–2323.

Tian, Jing et al. "Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with K-nearest neighbor distance analysis". In: *IEEE Transactions on Industrial Electronics* 63.3 (2015), pp. 1793–1803.

Tian, Zhigang. "An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring". In: *Journal of Intelligent Manufacturing* 23.2 (2012), pp. 227–237.

Tobon-Mejia, Diego Alejandro et al. "A data-driven failure prognostics method based on mixture of Gaussians hidden Markov models". In: *IEEE Transactions on reliability* 61.2 (2012), pp. 491–503.

Toubakh, Houari and Moamar Sayed-Mouchaweh. "Hybrid dynamic classifier for drift-like fault diagnosis in a class of hybrid dynamic systems: Application to wind turbine converters". In: *Neurocomputing* 171 (2016), pp. 1496–1516.

Toubakh, Houari and Moamar Sayed-Mouchaweh. “Hybrid dynamic data-driven approach for drift-like fault detection in wind turbines”. In: *Evolving Systems* 6.2 (2015), pp. 115–129.

Uluyol, Onder et al. “Power curve analytic for wind turbine performance monitoring and prognostics”. In: *Annual conference of the prognostics and health management society*. Vol. 2. 2011, pp. 1–8.

Vapnik, Vladimir. *The nature of statistical learning theory*. Springer science & business media, 2013.

Wang, Jiujian et al. “Remaining useful life estimation in prognostics using deep bidirectional lstm neural network”. In: *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*. IEEE. 2018, pp. 1037–1042.

Wang, Tianyi et al. “A similarity-based prognostics approach for remaining useful life estimation of engineered systems”. In: *2008 international conference on prognostics and health management*. IEEE. 2008, pp. 1–6.

Wang, Wilson Q, M Farid Golnaraghi, and Fathy Ismail. “Prognosis of machine health condition using neuro-fuzzy systems”. In: *Mechanical Systems and Signal Processing* 18.4 (2004), pp. 813–831.

Wang, Yu et al. “A two-stage data-driven-based prognostic approach for bearing degradation problem”. In: *IEEE Transactions on Industrial Informatics* 12.3 (2016), pp. 924–932.

Wu, Yuting et al. “Remaining useful life estimation of engineered systems using vanilla LSTM neural networks”. In: *Neurocomputing* 275 (2018), pp. 167–179.

Xia, Tangbin et al. “An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation”. In: *Computers in Industry* 115 (2020), p. 103182.

Yan, Jihong, Muammer Koc, and Jay Lee. “A prognostic algorithm for machine performance assessment and its application”. In: *Production Planning & Control* 15.8 (2004), pp. 796–801.

Yan, Weizhong and Lijie Yu. "On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach". In: *arXiv preprint arXiv:1908.09238* (2019).

Yin, Shen, Xiangping Zhu, and Chen Jing. "Fault detection based on a robust one class support vector machine". In: *Neurocomputing* 145 (2014), pp. 263–268.

Zhai, Lian-Yin et al. "Analysis of time-to-failure data with Weibull model in product life cycle management". In: *Re-engineering manufacturing for sustainability*. Springer, 2013, pp. 699–703.

Zhang, Ansi et al. "Transfer learning with deep recurrent neural networks for remaining useful life estimation". In: *Applied Sciences* 8.12 (2018), p. 2416.

Zhang, Bin, Lijun Zhang, and Jinwu Xu. "Degradation feature selection for remaining useful life prediction of rolling element bearings". In: *Quality and Reliability Engineering International* 32.2 (2016), pp. 547–554.

Zhang, Bin, Shaohui Zhang, and Weihua Li. "Bearing performance degradation assessment using long short-term memory recurrent network". In: *Computers in Industry* 106 (2019), pp. 14–29.

Zhang, Jianjing et al. "Long short-term memory for machine remaining life prediction". In: *Journal of manufacturing systems* 48 (2018), pp. 78–86.

Zhang, Yongzhi et al. "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries". In: *IEEE Transactions on Vehicular Technology* 67.7 (2018), pp. 5695–5705.

Zhang, Zhenyou, Yi Wang, and Kesheng Wang. "Fault diagnosis and prognosis using wavelet packet decomposition, Fourier transform and artificial neural network". In: *Journal of Intelligent Manufacturing* 24.6 (2013), pp. 1213–1227.

Zheng, Caifeng et al. "A data-driven approach for remaining useful life prediction of aircraft engines". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2018, pp. 184–189.

Zheng, Shuai et al. "Long short-term memory network for remaining useful life estimation". In: *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE. 2017, pp. 88–95.

Zhu, Junda et al. “Lubrication oil condition monitoring and remaining useful life prediction with particle filtering”. In: *International Journal of Prognostics and Health Management* 4 (2013), pp. 124–138.

Zhu, Zhi-Bo and Zhi-Huan Song. “A novel fault diagnosis system using pattern classification on kernel FDA subspace”. In: *Expert Systems with Applications* 38.6 (2011), pp. 6895–6905.

Zio, Enrico and Francesco Di Maio. “A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system”. In: *Reliability Engineering & System Safety* 95.1 (2010), pp. 49–57.

IMT Lille Douai
École Mines-Télécom
IMT-Université de Lille

Université de Lille

**Résumé :** Le pronostic des pannes consiste en l'estimation de la durée de vie résiduelle (RUL). Il s'agit d'un élément essentiel de la stratégie de maintenance prédictive qui peut contribuer à améliorer la fiabilité et la disponibilité des systèmes industriels tout en réduisant les temps d'arrêt imprévus et les coûts de maintenance. Fréquemment, l'application du pronostic dans des systèmes industriels réels est ardue en raison de l'insuffisance des données historiques de dégradation, en particulier pour les nouvelles machines, ou en raison du coût élevé de leur production ou de leur obtention. Par conséquent, ce manuscrit de thèse propose une approche générale axée sur les données avec trois chemins pour l'estimation du RUL lorsqu'aucune séquence de dégradation, peu de séquences et multiples séquences sont disponibles a priori. L'approche générale proposée comprend trois chemins d'estimation RUL : chemins aveugle, guidé et profond en fonction du nombre de séquences de dégradation disponibles. Le chemin aveugle est déclenché lorsqu'aucune séquence a priori n'est disponible, il vise à sélectionner le meilleur indicateur de santé (HI) dynamiquement à l'aide d'un critère de sélection basé sur le fit avec le modèle linéaire généralisé (GLM). Dans le même temps, les paramètres du GLM sont mis à jour en fonction des nouvelles données de dégradation permettant d'améliorer l'extrapolation de l'indicateur de santé jusqu'au seuil de défaillance. La prédiction du RUL s'améliore au fil du temps avec l'arrivée de nouvelles données de dégradation. Le chemin guidé est déclenché lorsqu'une ou plusieurs séquences a priori sont disponibles. Différents RUL sont estimés à l'aide de différents modèles adaptatifs combinés avec différents HI, tandis que le RUL final est obtenue en fusionnant les RUL en utilisant une moyenne pondérée. Le chemin guidé a montré de meilleurs résultats de prédiction du RUL par rapport aux méthodes appliquées dans la littérature (support vector machine et long short term memory). En outre, la précision et la robustesse de l'estimation RUL sont améliorées lorsque davantage de séquences a priori sont collectées. De plus, cette approche a montré une adaptation à plusieurs dynamiques (vitesses) de dégradation grâce à l'utilisation de différents modèles adaptatifs. Le chemin profond est utilisé lorsque de nombreuses séquences de dégradation a priori sont disponibles. Deux RULs sont prédits à l'aide de deux modèles d'apprentissage profond appelés Convolutional Neural Network (CNN) et Long Short Term Memory (LSTM), où le RUL final est obtenu en fusionnant les deux RUL en utilisant une moyenne pondérée. L'approche d'ensemble profond proposée a la capacité de capturer la variabilité des différents modes de conditions de fonctionnement. Cette approche a montré des résultats prometteurs par rapport aux méthodes similaires dans l'état de l'art.

**Mots clés :** Indicateur de santé - Détection de défauts - Pronostic des pannes - Durée de vie résiduelle

**Abstract:** Fault prognostics consists of the estimation of the remaining useful life (RUL). It is the main part of the predictive maintenance strategy that can help to enhance the reliability and availability of industrial systems while reducing unscheduled downtime and maintenance cost. Applying prognostics in real industrial systems is arduous due to the insufficiency of the historical degradation data, in particular for new machines, or because of the high cost to produce or obtain them. Hence, this PhD thesis proposes a general data-driven approach with three paths for the RUL estimation when no degradation sequences, few sequences, and multiple sequences are available a priori. The general proposed approach includes three RUL estimation paths: blind, informed, and deep paths depending on the number of available degradation sequences. The blind path is triggered when no a priori sequences are available, it aims to select the best Health Indicator (HI) dynamically using a selection criterion based on the goodness of fit with the Generalized Linear Model (GLM). The GLM's parameters are updated using the incoming degradation data in order to increase the RUL's accuracy estimation when extrapolating the HI until the failure threshold. Therefore, the RUL prediction is improved over time with the arrival of new incoming degradation data. The informed path is triggered when one or more a priori sequences are available. Different RULs are estimated using different adaptive models (GLM, quadratic regression, exponential model, and double exponential model) combined with different HIs, while the final RUL is obtained by merging the individual RULs using a weighted mean. The informed path showed better RUL prediction results when few a priori sequences are available compared with the State-of-the-art RUL estimation methods (support vector machine and long short term memory models). The RUL estimation accuracy and robustness are improved when more a priori sequences are collected. Besides. The informed path can predict the RUL when the degradation speed (dynamics) of the incoming sequence is different from the a priori degradation speed, or dynamics, thanks to the use of different adaptive models. The deep ensemble approach is used when many a priori degradation sequences are available. Two RULs are predicted using two deep learning models named Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM), where the final RUL is obtained by merging the two RUL using weighted mean. The proposed deep ensemble approach (deep path) has the ability to capture the variability of different operating condition modes. The deep ensemble approach showed promising results against the related works in the state-of-the-art.

**Keywords:** Health indicator - Fault detection - Fault prognostics - Remaining useful life