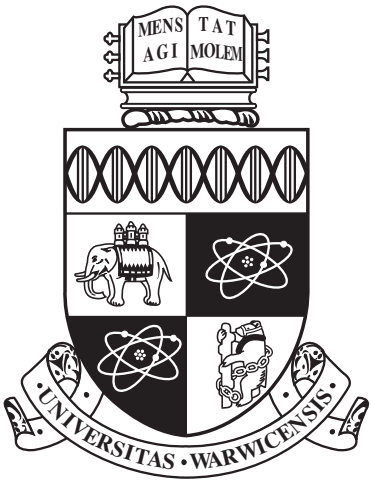

An Adaptive Data Filtering Model for Remaining Useful Life Estimation

by

Oguz BEKTAS

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Warwick Manufacturing Group

June 2018

THE UNIVERSITY OF
WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank my supervisors, Dr Jeffrey Jones and Dr Jane Marshall, for their kind support and guidance throughout my research. Their supervision and advice are of inestimable value for my study.

I want to express my thanks to my thesis examiners: Dr Faris Elasha and Dr James Marco for their kind comments and corrections in my thesis. I appreciate them taking the time out to examine my thesis.

I would like to thank Dr Kai Goebel and the Prognostics Center of Excellence of NASA Ames Research Center for their support of my work. Their hospitality during my research visit is most gratefully acknowledged.

I would also like to thank my department, research fellows, advisors and collaborators; their discussion, ideas, and feedback have been absolutely invaluable.

I would like to thank the people of Turkey for providing the government scholarship which allowed me the opportunity to fulfil my dreams.

Finally, I would especially like to thank my amazing family for their love, support, and encouragement. In particular, I would like to thank my wonderful wife "Aysenur Bektas" who has made countless sacrifices to help us get to this point and my little son "Yavuz Selim" who always gave his father extra motivation to finish this thesis as soon as possible.

To all those cited by name here and the many more I could not mention by their name, I will forever owe you this achievement.

Thank you all most sincerely - *Hepinize en içten dileklerimle teşekkür ederim*

Declarations

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Where other sources of information have been used, they have been acknowledged.

I undertake not to quote or make use of any information from this thesis without making acknowledgement to the author. I further undertake to allow no-one else to use this thesis while it is in my care.

The material is based upon work supported in part by NASA under award NNX12AK33A with the Universities Space Research Association. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Aeronautics and Space Administration.

Parts of this thesis have been published by the author:

1. Poster presentation at WMG Research and Innovation Conference on the 10th and 11th July 2014. *A Framework for Performance Metrics of Rotating Machinery Prognostics within Maintenance Free Operation Period -Oguz Bektas*

2. Conference Proceeding in The Twelfth International Conference on Condition Monitoring and Machinery Failure Prevention Technologies CM2015 / MFPT2015. *A Degradation Prognostic Framework for Gas Turbine*

Engines -Oguz Bektas & Jeffrey Jones

3. Technical Presentation at WMG Research and Innovation Conference on the 30th June and 11th July 2015. *Narx Model in Gas Turbine Prognostics -Oguz Bektas*

4. Conference Proceeding in Third European Conference of the PHM Society - PHME16. *NARX Time Series Model for Remaining Useful Life Estimation of Gas Turbine Engines -Oguz Bektas & Jeffrey Jones*

5. A journal article in Journal of Failure Analysis and Prevention. "*Reducing Dimensionality of Multi-regime Data for Failure Prognostics - Oguz Bektas, Amjad Alfudail, Jeffrey A. Jones*

Oguz BEKTAS **June 2018**

Abstract

The field of Prognostics and Health Management is becoming ever more important in the modern maintenance era, with advanced techniques of automation and mechanisation becoming increasingly prevalent. Prognostic technology has promising abilities to forecast remaining useful life and likely future circumstances of complex systems. However, the evolution of data processing and its critical impact on remaining useful life predictions continually demand increasing development so as to meet higher performance levels. There is often a gap between the adequacy of prognostic pre-processing and the prediction methods. One way to reduce this gap would be to design an adaptive data processing method that can filter multidimensional condition monitoring data by incorporating useful information from multiple data sources.

Due to the incomplete understanding on the multi-dimensional failure mechanisms and the collaboration between data sources, current prognostic methods lack the ability to deal effectively with complicated interdependency, multidimensional condition monitoring information and noisy data. Further conventional methods are unable to deal with these efficiently. The methodology proposed in this research handles these deficiencies by introducing a prognostic framework that allows the effective use of monitoring data from different resources to predict the lifetime of systems. The methodology presents a feed-forward neural network filtering approach for trajectory similarity based remaining useful life predictions. The extraction of health indicators is applied as a type of dynamic filtering, in which the time series having full operational conditions are used to train a neural network mapping between raw training inputs and a health indicator output. This trained network function is evaluated by repeating condition monitoring information from multiple data subsets. After the network filtering, the training trajectories are used as baselines to predict the future behaviours of test trajectories. The similarity between these data subsets compares the relationship between the historical performance deterioration of a system's prior operating period with a similar system's degradation behaviour.

The proposed prognostic technique, together with dynamic data filtering and remaining useful estimation, holds the promise of increased prediction performance levels. The presented methodology was tested using the PHM08 data challenge provided by the Prognostics Centre of Excellence at NASA Ames Research Centre, and it achieved the overall leading score in the published literature.

Keywords: *Prognostics and Health Management, Remaining Useful Life, Data Driven Prognostics, Artificial Neural Networks, Similarity-Based Predictions*

Symbols

The following list contains the symbols most frequently used in this thesis. References to sections are given at the beginning and then the symbols are defined.

Section 2.1 & 2.2

$\mathbf{\Omega}$	Dimensions space
$\mathbf{\Psi}$	Dimension levels space
$\mathbf{C_b}$	Multidimensional data set
D	List of dimensions
L	List of regime levels
R	Set of multidimensional data
x	Multidimensional data
H	Health Measure
$\Delta\theta$	Life distance
ε	Noise
$p_G(\varepsilon)$	Gaussian Noise
μ	Mean
σ	Standard Deviation
SNR	Signal to Noise Ratio
P_{signal}	Power of signal
P_{noise}	Power of noise
X	Variable of RUL
t	Time index
Z	Preceding condition profile
R_t	Survival function
θ_t	History of operational profiles
PH	Prognostic horizon
EOP	End of prediction

Section 2.3

a	Crack length
ΔK	Range of stress intensity factor
q	Training trajectory
p	Test Trajectory
f_s	State transition function
f_m	Measurement function
$\ddot{x}$	Damage State
$\ddot{p}$	Model parameters
$\dot{\varepsilon}$	Process noise
$\ddot{\varepsilon}$	Measurement noise

$\boldsymbol{Fs}$ State evolution matrix

$\boldsymbol{Fm}$ Measurement matrix

P Autoregressive model order

Q Moving average model order

ϕ Auto-regressive term

Θ Moving average term

y Target variable

β Coefficients

d Regime

h Health index

$e(t)$ Efficiency

H Overall health index

w Weight

b Bias

M ANN model

$\mathbf{H}$ Hessian matrix

Section 2.4

e Error

AE Absolute error

MAE Mean absolute error

AB Average bias

MSE Mean square error

FP False positive rate

FN False negative rate

$MAPE$ Mean absolute percentage error

MAD Mean absolute deviation from the sample median

$MdAD$ Median absolute deviation from the sample median

s Score function

Section 3

$\mathbf{\Omega}$ Multidimensional space

$\mathbf{T}$ Operational trajectory

$\mathbf{r}$ Regimes (operating conditions)

$\mathbf{x}$ Raw condition monitoring data

n Number of time series points

$\theta(t)$ Number of unique time index variables

$\theta(r)$ Number of unique regimes

np Number of regimes

l_r Lower regime domain limits

u_r Upper regime domain limits

ϑ_r Regime domain

w^p Noise and measurement uncertainties at regime, p

H^p Health information at regime, p

$t_{current}$ Time stamp of the current measurement index

t_{end} Ending time stamp

t_{test} Time index of the test instance

$t_{training}$ Time index of the training (run-to-failure) instance

h Health level

d Initial wear level

a, b Degradation model parameters

ns Number of sensors

f_{NN} Neural network function

y Neural network output (target)

$N(x)$ Normalised data

x^r Raw data at regime r

σ^r Standard deviation of regime r

μ^r Mean of regime r

c Cluster variable

$x_{(c)}$ Raw data at cluster c

$\sigma_{(c)}$ Standard deviation of cluster c

$\mu_{(c)}$ Mean of cluster c

$\hat{x}_{(c)}$ Normalised data at cluster c

s Moving average

$f(\mathbf{x}_{(t)})$ Feed forward neural network function

w_i Fixed real-valued weights

$L(i)$ Neural network library

l Number of trained network functions

d Euclidean distance

n_{tr} Length of the base curve (training trajectory)

n_{te} Length of test trajectory

Mn Stored pairwise distances

$L_{te,tr}$ Location of the best-matching part at the training baseline

Section 4

sHI Synthetic health indicator

Abbreviations

AB	Average Bias
ANN	Artificial Neural Network
$ARMA$	Auto-Regressive Moving-Average
$ATTF$	Actual Time to Failure
BR	Bayesian regularisation
CBM	Condition Based Maintenance
$C-MAPSS$	Commercial Modular Aero-Propulsion System Simulation
ES	Expert Systems
EOP	End of Prediction
$ETTF$	Estimated Time to Failure
FEA	Finite Element Analysis
FL	Fuzzy Logic
FN	False Negative Rate
FP	False Positive Rate
HI	Health Indicator
HPC	High Pressure Compressor
HPT	High Pressure Turbine
HMM	Hidden Markov Model
$IGBT$	Insulated Gate Bipolar Transistors
KbM	Knowledge-based Models
KF	Kalman filtering

LPC	Low Pressure Compressor
LPT	Low Pressure Turbine
LQE	Linear Quadratic Estimation
MA	Moving Average
MAD	Mean Absolute Deviation from the Sample Median
$MdAD$	Median Absolute Deviation from the Sample Median
MAE	Mean Absolute Error
$MAPE$	Mean Absolute Percentage Error
MLP	Multi-Layer Perceptron
MRO	Maintenance, Repair, and Operations
MSE	Mean Square Error
NN	Neural Network
$NFLT$	No Free Lunch Theorem
PbM	Physics-based Models
PCA	Principal Component Analysis
$PCoE$	Prognostics Center of Excellence
PDF	Probability Density Function
PF	Particle filtering
PH	Prognostic Horizon
PHM	Prognostics and Health Management
PM	Preventive Maintenance
$RBFN$	Radial Basis Function Networks
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
SD	Secondary Dataset
SMC	Sequential Monte Carlo
SNR	Signal-to-Noise Ratio

Chapter 1

Introduction

The focus of reliable systems community is on the basic principles of system failures to explain how complex systems age and fail. Over the last decade, Prognostics and Health Management (PHM) has been an emerging discipline as a responding point to this focus, and it has established a connection with failure mechanisms and system life-cycle management (Uckun et al., 2008). PHM aims to maximise operational availability and safety by incorporating functions of condition monitoring, prognostics, state assessment, diagnostics and failure progression (Sheppard et al., 2008). Due to the constantly increasing interest in PHM technologies, which has received great attention from the industry at all levels, there has been a significant change in attitude towards these functions.

As a steadily growing subject in PHM applications, prognostics are a particularly well-known practice in a wide variety of applications (Lee et al., 2014). The term "prognostics" in the context of PHM field refers to the estimation of a time at which a system (or a component) is at the end of its useful life point and will no longer perform its desired functional requirements (Saxena et al., 2014). Therefore, prognostic methods are mainly focused on remaining useful life (RUL) predictions in terms that are beneficial to the main-

tenance decision making process (Sandborn and Wilkinson, 2007). Within the last decade, these methods have advanced expertise in various disciplines and an overall understanding of the predictions for health management has substantially improved. Many breakthroughs in prognostics and their ability to predict RUL for maintenance purposes can be found in various areas, notably in advanced engineering systems such as, gearboxes (Elasha et al., 2014a,c, 2015b,c, 2017), batteries (Pastor-Fernández et al., 2016; Saha et al., 2009; Goebel et al., 2008a; Rezvani et al., 2011), actuators (Byington et al., 2004b), fuel cells (Morando et al., 2013), turbofan engines (Wang et al., 2008; Heimes, 2008; Peel, 2008) and even NASA's launch vehicles and spacecraft systems (Luchinsky et al., 2007).

Prognostic technology has also been used to design platforms with prediction capacity as an integral feature of the overall system architecture (Uckun et al., 2008), such as the Joint Strike Fighter Programme and the Future Combat Systems Programme (Hess et al., 2004; Barton, 2007). As the technology develops further, prognostics continue to play a prominent role in the evolution of such complex systems. Introducing prognostic development to the earliest stages of complex system design can highlight elements requiring reconsideration earlier in the design stage and allows the process of testing, validating, and refining to provide a higher quality design (Kramer and Tumer, 2009). Many advanced modelling techniques exist for system design and significant research have been related to the predictive capabilities for future operations, and resulted in the development of notable prognostic applications. The majority of these applications concentrate on imposing alternative strategies for maintenance, and have led to a huge and diverse literature on machinery prognostics with a considerable number of papers, substantial theories and practical models (Jardine et al., 2006a).

1.1 Motivation

Today, many sophisticated sensors and computerised components are capable of delivering system's performance data to prognostics that can continuously track health degradation and extrapolate the temporal behaviour of system's health to predict risks of unacceptable performance over time as well as synchronising necessary maintenance actions with the overall operation of the system (Lee et al., 2006). Since the key point in maintenance actions is the involvement of all operation-related activities (BSI, 1993), the prognostic applications are directly related to the development of prediction-based maintenance actions required to keep systems at a desired level of maintenance with minimum operational cost. The structure of maintenance, accordingly, is highly interrelated with prognostics to reduce maintenance expenses. To gain an appreciation for prognostic technology and amounts of maintenance expenditure, the global airlines market can be considered an illustrative example. In 2014, the worldwide airliner spending on maintenance, repair, and operations (MRO) accounted for $62.1 billion, in which engine maintenance was about 40% of the total cost (IATA, 2011). Although the size of the MRO market is expected to reach $90 billion in 2024, it is estimated that the developing trends in such areas as prognostics, innovations and technologies will reduce MRO costs by 15 to 20%, and predictive maintenance strategies are expected to increase airliner availability by up to 35% (IATA, 2011).

As the predictive maintenance strategies develop, the ability to find solutions to key challenges have become a crucial push towards developing models that apply performance in maintenance. In this context, data processing for maintenance decision making can be regarded a major step in a successful maintenance programme (Tsang et al., 1999; Jardine et al., 2006a). In order to avoid unnecessary maintenance tasks, data processing analyses available condition information on which maintenance decision making is re-

quired (Lee et al., 2014). However, unless the uncertainties in the predictions are also considered, data processing may not be a sufficiently practical step for decision makers to justify the need for prognostic actions (Goebel et al., 2013). Since it is very unlikely to attempt to estimate the operating and environmental conditions under which a given system operates, a systematic framework for data processing is necessary to account for the uncertainties in prognostics (Sankararaman and Goebel, 2015). Data processing should handle and analyse the condition monitoring data for a better understanding and interpretation of the system's damage propagation, which characterises how the damage is expected to grow in upcoming operational and environmental conditions, along with any other effects that might have an impact on damage (Goebel et al., 2013).

1.2 Problem Statement

Recent technological advances have resulted in increasingly complex systems that pose considerable challenges in terms of successful operations and use over their life cycles (Venkatasubramanian, 2005). Ensuring safety and performance in complex and safety-critical systems is a major problem to be tackled, and above all, complexity is one most prominent issue that must be addressed to make theoretical models applicable to real-life applications (Boussif, 2016). A complex engineering system is defined as a group of interrelated, interacting, and/ or interdependent components (constituents) forming a complex whole (Jamshidi, 2008).

Condition monitoring of such complex systems is usually based on various sensors of components to receive information on the system health status and recognise any potential problems at an early stage so that corrective maintenance actions can be taken in a timely manner. However, evaluation and

interpretation of sensor data from multiple elements is often a challenging aspect due to complicated interdependencies between monitored data and actual system conditions (Günel et al., 2013).

As a result of operational conditions and regimes, some complex system works under different superimposed operational margins at any given time instant and the wearing process of such systems is not usually deterministic, and commonly not one-dimensional (Saxena et al., 2008b). These multidimensional and noisy data streams are measured from a large number of monitoring channels from a population of similar components such as the operational or environmental conditions, direct and indirect measurements that are potentially related to the damage progress (Uckun et al., 2008). Therefore, a simple model is mostly unable to present the wearing phenomena and one should consider a more advanced decision-making process for the condition monitoring data in multidimensional form (Cempel, 2009).

While the literature has recognised the importance of the multiple axes of information and multidimensional data, there is still a lack of analysis in such data, leaving the analysts with yet more information and data to process through the complex systems (Tumer and Huff, 2003). This can be attributed to two fundamental issues:

- Incomplete information on the multi-dimensional mechanisms failure and fault modes;
- Lack of collaboration between different but similar data sources

When a prognostic model planned and tested under controlled experiments, it has to face the challenges brought up by the complexity of real-world systems (Wang, 2010). The models should reduce the future uncertainty of operations according to the needs of system itself, and limitations of condition monitoring data. In conventional applications, the selection of a suitable

prognostic algorithm for a successful and practical implementation in complex systems depends on understanding of the challenges associated with the type of applications (Zaidan et al., 2013). Therefore, the majority of prognostic researches to date has been mostly disparate, theoretical and restricted to a small number of cases, and there are few published examples of full prognostic frameworks being applied in the field of complex systems where the monitored condition data are exposed to a range of dynamic operating conditions (Sikorska et al., 2011).

1.3 Research Question and Hypotheses

The key element to design prognostics for such complex systems is the data processing for conditioning and feature extraction of acquired multidimensional data. For RUL estimations, data processing capabilities are of greater significance for design and implementation point of view (Saxena et al., 2008a). If condition monitoring data is not available sufficiently, the prognostic requirements cannot be met (Uckun et al., 2008). One way to satisfy the prognostic requirements is to perform a data collaboration effort distributed across multiple suppliers, rather than using separate and self-contained sources. Collaboration, from this perspective, is the act of working together with one or more sides in order to achieve the objective of higher accuracy (Soukhanov, 2001); when used in intelligence-intensive activities, this may lead to increased and improved results (Moyle, 2009).

Additionally, data processing can be understood as the conversion of multidimensional and complex monitoring data to meaningful information for RUL estimation. Because the effort needed to filter prognostic parameters from multiple data can quickly make the prognostic approach applicable for more applications, a collaborating data processing method which results in an

optimal, or near-optimal, filtering is very promising for RUL estimations.

The literature shows that because of the lack of understanding on the multi-dimensional failure mechanisms and incomplete collaboration between different but similar data sources, current prognostic methods are unable to deal effectively with complicated interdependency (Günel et al., 2013), multi-dimensional data (Saxena et al., 2008b) and noisy data (Uckun et al., 2008). The traditional pre-processing models lack the ability to address this efficiently. The method proposed in this research handles these issues by introducing an adaptive data filtering model for RUL estimation.

The main focus in the research is to make an original contribution to the development of prognostics by considering data filtering requirements on multiple data resources. The point at issue is the modelling of such a conceptual prognostic framework to overcome the challenges presented by different datasets and to increase prediction performance. The model should be based on adaptation of complicated prognostic filtering algorithms for different external input resources and satisfying the generic demands on the performance of RUL predictions by using alternative trained functions. In this respective, an effective method is necessary to approximate any filtering function arbitrarily well.

In this work, the intention to use such an adaptive filtering function is to visualise information from different but similar data sources, and get this information to the right place, in the right arrangement, and in good time to use it in making effective predictions. Then, a prognostic prediction method can be swamped with data from multiple sources and investigate data filtering and processing techniques, with the ultimate goal of improving RUL predictions.

Research Question

The fundamental research question pertaining to the scope of this study is:

"How can an adaptive data filtering model be designed as a part of multi-step ahead remaining useful life predictions? "

Research Aim

The aim of this research is to study the possibility of developing a prognostic and health management prediction approach to complex machinery systems under dynamic operating regimes. This prediction is based on the systematic monitoring of multidimensional data and the filtered available information, and it is in line with adaptable stages consisting of data processing and multi-step ahead predictions. Therefore, the research aims to develop a multifaceted prognostic approach that integrates data filtering and RUL estimation to enhance the prediction results and to increase the applicability of prognostics and health management in machinery applications under a set of operational cases with multi-regime operating conditions.

Research Objectives

The research has following key objectives that will help to satisfy the above-mentioned issues and address the current gap indicated by the research question.

- To develop an understanding of emerging concepts of prognostic use in maintenance
- To design an adaptive prognostic model that can process filtering and prediction modes
- To advance data filtering processes by structuring a training library in which health of various operational cases can be calculated

- To develop a merging process between data qualification and RUL estimation
- To increase the performance of RUL estimations and provide more detailed results in terms of maintenance planning
- To reduce excessive RUL prediction error rates for critical cases
- To minimise the gap between false positive or negative error rates of life-time predictions

1.4 Significance and Contribution

The significance of this study can be found in the sense that it can introduce a novel perspective to data filtering and prognostics accuracy by considering multi-dimensional condition monitoring data from complex systems. A consideration of operational requirements in terms of multiple data employment has improved the prediction performance of prognostics. An adaptive prognostic method is designed to introduce a sequential process from training to prediction. A neural network-based data training function is presented to map between a set of raw monitoring inputs and a set of targets. Then, external datasets are applied to this function to allow for an adjustment from their raw scales to a notionally common target scale. This allows for the fact that multiple data sources can be filtered separately and be used for RUL estimations.

The contribution of this study, as a response to the formulated research question, is the introduction of a data-driven filtering process (feed-forward neural network) into collaborative RULs (trajectory similarity-based prognostics). In particular, the research investigates how pre-processing methods affect algorithm performance. The algorithm will be tested by PHM08 data challenge provided by the Prognostics CoE at NASA Ames Research Centre.

1.5 Thesis Organisation

This thesis is composed of six chapters. The organisation of thesis is as follows:

Chapter 2 reviews the progress of the current literature and gives the state-of-art on current prognostic applications as well as the various sources and methods applied in complex system domains.

Chapter 3 includes the methodology sections. This chapter provides the details of the multi-regime normalisation, data filtering and RUL prediction frameworks. A feed-forward dynamic network relating to the raw input time series with a target vector is introduced. Then, the similarity-based forecasting setup is summarised.

Chapter 4 applies the presented method to different case studies which are then used as examples of how to gain a better understanding of the risks posed by various conditions. Case studies contribute to more focused analyses which, in the context of damage, demonstrate the effectiveness of prognostic methods and metrics, and identify risks in multi-step ahead projections.

Chapter 5 brings forward the analysis, performance evaluation and validation of the model, as based on true RUL parameters. The chapter also deals with the issues of critical assessment of this study, contribution to knowledge, and a comparison of this model with rival ones.

Chapter 6 presents further work, a summary and conclusions. It outlines the research and possible future work is made apparent.

Chapter 2

Background and Literature Survey

Maintenance has witnessed substantial changes, perhaps more so than any other discipline in the industry as a direct result of an increase in the variety and quantity of complex systems requiring novel practices and progressive views (Moubray, 1997). Due to the increasing awareness of high plant availability and reliability, maintenance strategies have been going through a stage where there is a necessity to respond the rapidly changing expectations. There is a significant risk that complex systems and their related interfaces would not be satisfied by classic breakdown repairs; therefore, various approaches in maintenance applications have been defined in an attempt to meet their increasing requirements (Jardine and Tsang, 2013).

A classification of maintenance is shown in Table 2.1. Simply, the policies are divided into unplanned and planned categories according to the historical perspective of maintenance. These types are related to how a user would like to approach repair procedures. Determination of the optimum schedule for performing maintenance has been a long-standing challenge. Maintenance strategies have evolved over time from run-to-failure, also known as breakdown

Table 2.1: Classification of Maintenance (BSI, 1993; Jones, 2010)

Unplanned Maintenance: or "run to failure", is an option of "do nothing until it breaks"

Corrective Maintenance	It is carried out after a failure has occurred
Emergency Maintenance	EM is required where immediate action is necessary to avoid serious consequences

Planned Maintenance: carried out and organised with forethought

Preventative Maintenance	PM performed at predetermined intervals. It aims to reduce the probability of failure.	
	Scheduled Maintenance	It attempts to forestall breakdown by operating on a predetermined interval of time
	Condition-Based Maintenance	CM is initiated as a result of condition monitoring knowledge. It seeks to determine actual operating conditions

maintenance, to preventive maintenance (PM) schemes, and then to condition-based maintenance (CBM) (Heng et al., 2009). Breakdown maintenance was the earliest attitude taken, including the repair or replacement of equipment when a failure occurs (Misra, 2008). This type of basic approach shifted towards PM in order to prevent unscheduled downtime and avoid catastrophic failures. However, each maintenance type has still its own unique advantages that make each desirable for specific needs. For example, until the most spectacular changes that occurred in maintenance after World War II (Brown and Sondalini, 2014), corrective maintenance had been the only option for a maintainer to fix or replace an item; that is after a failure had occurred. Nevertheless, corrective maintenance is still in use for simple components in which the failure consequences are less critical. Moreover, no matter how extensive a maintenance programme is, there is always the possibility that critical systems

may fail and corrective maintenance become necessary (Sheut and Krajewski, 1994). On the other hand, changing customer requirements demand planned maintenance designs at all levels. From the 1950s, mechanisation and automation of systems have arisen due to an increasing intolerance of downtime and labour costs, with the first planned maintenance practices introduced in the form of preventive maintenance to avoid catastrophic failures and emergency shutdowns (Heng, 2009).

In the case of PM, maintenance is carried out on a regular basis to reduce the failure rate or performance decrease of given piece of equipment (BSI, 1993). Preventive actions involve fixed scheduled servicing, inspections, and repairs to reduce failures and costs. In such a procedure, maintenance is considered where inspections and overhauls take place at different time intervals without seriously considering the condition of a given system's health (Badıa et al., 2002).

Bazovsky (1961) laid the groundwork for the application of mathematical optimisation methods in PM plans and Jardine (1973) pioneered decision models for assignment of optimal replacement and/or repair intervals by investigating historical breakdown measures and cost implications. In subsequent years, Lee and Rosenblatt (1987) analysed the use of machine inspection in PM for restoration purposes of economical manufacturing quantity, and Groenevelt et al. (1992) studied the role of PM in an unreliable production system with a constant failure rate and randomly distributed repair times. Other studies have also shown that PM strategies can be an effective way to extend the lifetime of several randomly failing production units and reduce operational costs (Barlow and Proschan, 1996; Nakagawa, 1981; Nakagawa and Yasui, 1991). Under the assumption of non-negligible maintenance restoration times, all these studies addressed the use of planned maintenance that is regularly performed on a system to lessen the possibility of failure. However, as a result of the increas-

ing system complexity, these scheduled maintenance policies have financial and safety implications that may result in labour-intensive and unnecessary maintenance actions that can be associated with over-maintaining. While PM strategies are robust enough to continue earlier and relatively simple routine maintenance tasks to reduce component damage, they may have shortcomings when dealing with catastrophic failures in complex systems. PM approach can be costly when it is, or needs to be, performed frequently (Bohlin et al., 2010), and fixed scheduled maintenance policies are not considered functional by most practitioners (Kelly, 1989). This is why, it is necessary to invest in monitoring efforts to maintain the correct item at the right time. Dynamic planning using malfunction signs is required to reduce unpredictable events and unnecessary maintenance tasks.

In recent times, the performance of PM has started to be questioned. Wood (1999) and Moubray (1997) defended the idea that scheduled and time-based PM often fails to make best use of the remaining service life; on many occasions, units are replaced despite the fact that they have many hours of RUL. The fixed repair intervals required by PM tasks waste a considerable amount of resources (Tsang, 1995), such that systems or units are often unreasonably maintained (Ellis and Byron, 2008). These queries resulted in the subdivision of CBM.

A properly and effectively established CBM programme can notably decrease maintenance costs by dropping unnecessarily scheduled preventive maintenance actions (Jardine et al., 2006a). CBM is similar to PM in the sense that it aims to prevent abnormality in advance of an incident, but it differs from the fixed time-oriented approach of scheduled maintenance (Shin and Jun, 2015). Hence, CBM offers an alternative to PM considerations of simple age-related failure modes.

The reduced time-based maintenance intervals of CBM are drawing at-

tention from maintenance service providers and end users who require higher planned maintenance performance, and the reduction of unnecessary inspections (Chen et al., 2012; Ahmad and Kamaruddin, 2012; Gulati and Mackey, 2003). Use of CBM in industry, therefore, has been reported to be one of the most substantial ways to decrease maintenance budgets (Bengtsson, 2004), and have thus seen a considerable increase in their uptake over time. For example, in 1981, domestic plants in the United States had spent more than $600 billion in maintaining critical systems; by 1991, this cost increased to more than $800 billion, and exceeded $1.2 Trillion in 2000 (PlantServices, 2004). Considering that the cost of ineffective maintenance constitutes around one-third of the total maintenance cost, and there is a similar trend in many other countries, there is a clear and immediate need to continuously progress and improve current maintenance strategies (Heng et al., 2009). As a response to this need, CBM allows maintenance service providers to avoid the inevitability of high-cost maintenance due to the elimination of redundant maintenance activities (Shin and Jun, 2015).

2.1 Condition Based Maintenance

Condition-Based Maintenance is a programme recommending maintenance actions based on condition monitoring data (Raheja et al., 2000); it attempts to avoid redundant tasks by taking repair actions only when there is evidence of unexpected operational issues (Jardine et al., 2006a). Certain signs, condition changes, or malfunction indications can precede the vast majority of machine failures (Pusey, 1999; Heinz P. Bloch, 2012), and maintenance actions based on monitoring can be taken prior to a severe performance reduction in a given system, with condition-based monitoring offering the ability of yielding accurate prediction results for mechanical failures.

According to Horner et al. (1997), the optimal time period for performing maintenance could be decided by the actual monitoring of a given system, and its subcomponents or units, and the condition monitoring assessment could vary from basic visual checks to detailed automatic inspections. Considering the continuous data related to the operational condition of critical systems, Knapp and Wang (1992) argued that CBM aims to minimise the cost of repairs, and Knapp et al. (2000) noted that condition monitoring can provide adequate warnings as to pending failures, which would thus allow for more detailed planned maintenance actions as based on system degradation.

On the other hand, as stated by Ellis and Byron (2008), CBM in some instances may not be cost-effective or sufficient data may not exist to justify intervention. Raheja et al. (2006) also claimed that some current CBM approaches are extremely specific and a common structure for CBM is missing, so that each domain has its own understanding that may not be well-suited for the requirements of other applications. Although CBM may not be applicable to all maintenance systems, it could be applied in instances where modules of condition-monitoring are available and are well-integrated (Horner et al., 1997)

Such typical modules of effective CBM implementation include the stages of data acquisition (information collecting), data processing (information handling) and maintenance decision-making (see Figure 2.1). These three key elements provide the necessary conditional information on which the maintenance process can be based, and help to avoid unnecessary maintenance operations. The data acquisition and processing modules in a CBM programme are critical pre-processing stages that present and process monitoring data to produce the information useful to the diagnostic and prognostic stages. After the raw condition monitoring data is processed, organised and restructured, diagnostics deal with fault and off-nominal condition detection, isolation, and

identification, while prognostics deal with the prediction of the future lifetime of a system over a fixed time horizon (Brotherton et al., 2000). These lifetime estimation applications are key enablers of the CBM strategy (Byington et al., 2008), and have significant value within complex system operations in terms of maintenance scheduling thus the reduction of maintenance downtime and costs (Brotherton et al., 2000).

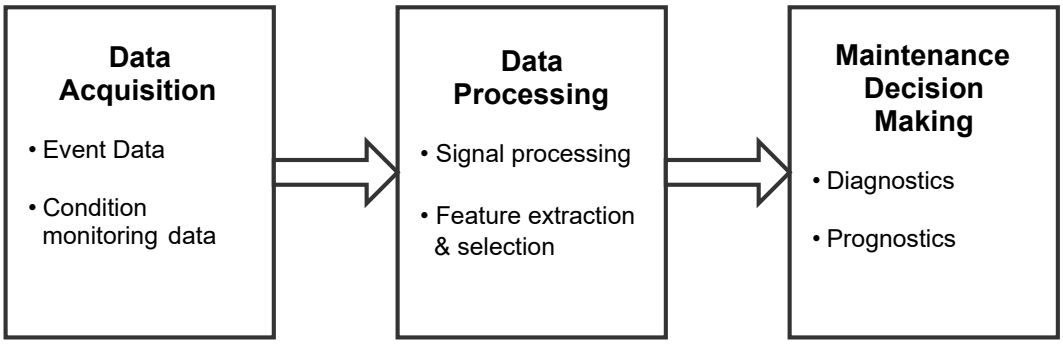

Figure 2.1: Major Steps in CBM

CBM objectives are designed to determine the condition and the remaining life of in-service equipment. Diagnostics and/or prognostics can be effectively used according to these objectives, and applied when developing the decision-making stage of complex maintenance applications (Heng et al., 2009). They are considered as significant procedures effecting a substantial shift in the current manner of technology that can push the boundary of systems health management (Elattar et al., 2016).

2.1.1 Data Acquisition

In the data acquisition stage, useful information is collected and stored from target physical assets in order to monitor the condition of systems, diagnose faults, and predict the remaining lifetime of assets (Tran and Yang, 2012). In parallel with the increasing development of advanced computer and sensor systems, data acquisition facilities have become more powerful and less expen-

sive technologies that could implement data for CBM in a more affordable and feasible way (Jardine et al., 2006a).

Data acquisition for a CBM programme is mainly supported by condition-monitoring data. For a complex mechanical system, condition monitoring information can be very diverse; superficially, it can constitute temperature, acoustic, oil analysis, environment moisture or weather data, etc. Condition monitoring of such systems in operation is generally multidimensional (Cempel, 2003). The analysis and extraction of useful information from multidimensional data is inevitably based on high-performance computing capabilities of data processing (Baurle and Gaffney, 2008).

Although condition monitoring could provide sufficient information for the development of data processing, there is still the requirement to collect common datasets and a mutual comparison to validate the methods introduced by different researchers. However, the nature of data acquisition has its own challenges in terms of the availability of condition monitoring data (Eker et al., 2012). To start the acquisition of raw monitoring data for effective lifetime estimations, the first major issue is to find available data sources. A common database is an important aspect for life estimation applications of maintenance decision making (Kans and Ingwald, 2008). Considering that such a database includes information from several but similar engineering domains, it can form the basis for the stage of data acquisition and required actions (Simões et al., 2011).

Use of the common database practice allows for the comparison of current applications and potential areas of improvement with regard to other researches in the literature (Bask et al., 2008), and since this type of a database can provide easy access to relevant data, it can demonstrate the detection of deviations at an early stage (Simões et al., 2011). However, Saxena et al. (2008b) stated that the data acquisition stage faces the perennial challenge of

a deficiency of common run-to-failure data sets, and that real-world data in most cases includes fault signs for an increasing degradation but no or little data capture for fault progress until failure. There have been synthetic data generating methods that can closely match the characteristics of raw data such as Eklund (2006); Zaidan et al. (2013); Bektas and Jones (2015) but these methods were developed for individual cases and the common data sets through which researchers can compare their approaches are required for an effective data acquisition stage in CBM.

2.1.2 Data Processing

A major issue confronting condition monitoring is ability to handle the collected data in terms of measurable machinery conditions. Data processing is the most critical stage in CBM because it is not always clear during the collection and manipulation of raw data from different sensors when meaningful information useful to lifetime estimations is being produced (Vatani, 2013).

Condition monitoring data collected from the data acquisition stage is versatile and falls into various categories such as value type (single value collected at a specific time epoch), waveform type (time series of a condition monitoring variable) and multidimension type (multidimensional time series acquired from multiple operating conditions) (Jardine et al., 2006b) .

In the case of complex systems under dynamic operating regimes, special attention is given to multidimension type data as it requires more processing and many techniques have been developed for its analysis and interpretation. Raw multidimensional data acquired from sensors providing information under multiple operating conditions are almost always processed before it is used for further analysis.

A typical complex system is made of different interacting components in which the collective actions are hard to deduce from those of the individ-

ual components, system predictability is limited, and the responses do not scale linearly (Bar-Yam, 2003). Additionally, the technological advancements have resulted in easy and efficient generation of data sets, most of which are multidimensional and/or multivariate (Seo, 2005).

A multidimensional database is composed of sets of vectors on relational elements constructed from hierarchies of dimension levels (Vassiliadis, 1998). Considering $\mathbf{\Omega}$ as the space of all dimensions and $\mathbf{\Psi}$ as the space of all dimension levels, for each dimension D there exist a set of regime levels and there is a set of values belonging to it. Accordingly, a basic multidimensional data set, $\mathbf{C_b}$, can be defined as a 3-tuple $< \mathbf{D_b}, \mathbf{L_b}, \mathbf{R_b} >$ (Vassiliadis, 1998), where

$$\mathbf{D_b} =< D_1, D_2, \cdots D_n, M > \tag{2.1}$$

is a list of dimensions ($D_i, M \in \mathbf{\Omega}$) and M is a dimension that represents the measure of C_b.

$$\mathbf{L_b} =< DL_{b1}, DL_{b2}, \cdots DL_{bn},^* ML > \tag{2.2}$$

is a list of regime levels ($DL_{bi},^* ML \in \mathbf{\Psi}$). ML is the multi-valued dimension level of the measure of C_b. $\mathbf{R_b}$ is a set of cell data and formed of

$$x = [x_1, x_2, \cdots, x_n] \tag{2.3}$$

Having specified this cell data, there is a possibility to assess the system breakdown time, θ_b, by using the life dependence of observed sensor readings, R, in terms of a function of some simplified health measure, H (Cempel, 2003).

$$H = \frac{t}{t_b}, \quad 0 \leq H < 1 \tag{2.4}$$

where t is the current system life time and t_b is the anticipated breakdown time. A single observed sensor of order n points dependent on the health measure, H, is expressed as:

$$R_n(t) = R_n\left(\frac{t}{t_b} \cdot \theta\right) = R_n(H \cdot t), \tag{2.5}$$

$$n = 1, 2, \cdots, r \tag{2.6}$$

Usually, the signal observations are over some life distance, $\Delta\theta$, in which the condition of the operating system, hence the values of observed signals, may substantially change (Cempel, 2003).

$$R_n(t_m) = R_n(m\Delta t) = R_{mn}, \tag{2.7}$$

$$n = 1, 2, \cdots, r, \quad m = 1, 2, \cdots, p \tag{2.8}$$

Condition monitoring data in such a multidimensional and multivariate form needs to be processed to produce meaningful information. Appropriate data characteristics are required to be calculated, selected and/or extracted for effective life estimations of the monitored systems (Heng, 2009). In this respect, data processing module carries out the functions of signal processing, feature extraction and selection (Tran and Yang, 2012). The data in these steps is processed to remove distortions and re-establish the actual form of signals.

Raw signals received from sensors are also generally very noisy, have very low signal-to-noise ratios and can be biased. Referring to equation 2.5, the noise in data needs to be added to the single observed sensor dependent on the health measure.

$$R_n(t) = R_n(H \cdot t) + \varepsilon(t), \quad 0 < t \leq t_b \tag{2.9}$$

where ε is a random variable accounting for noise and/or the measurement uncertainties. Since the values in the term ε are identically distributed and statistically independent (and hence uncorrelated), it is commonly assumed that ε is drawn from a Gaussian white noise (It is called white in analogy to white light which has uniform emissions at all frequencies in the visible spectrum) (Li et al., 2000; Menezes and Barreto, 2008; Lam et al., 2014). The probability density function (p) of such a Gaussian noise (ε) is given by the normal distribution.

$$p_G(\varepsilon) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\varepsilon-\mu)^2}{2\sigma^2}} \tag{2.10}$$

where μ and σ represent the mean value and the standard deviation respectively (Cattin, 2013). The relative power of ε in the channel of R_n is typically described by the measure of "signal-to-noise ratio" (SNR) per sample which is defined as the ratio of the power of a signal to the unwanted signal (Ede et al., 2010).

$$\text{SNR} = \frac{P_{signal}}{P_{noise}} \tag{2.11}$$

where P is average power. Both the power of signals and noise are measured at the same points, and within the same bandwidth.

To extract useful information from the data where the ε is high and SNR is low, various signal processing methods have been developed to evaluate and analyse the characteristics of noisy and multidimensional raw data (Jardine et al., 2006a).

The common techniques used those of Fourier Transform (Schoen and Habetler, 1993; De Almeida et al., 2002; Liu et al., 2004) and Wavelets Transform (Staszewski and Tomlinson, 1994; Wang and McFadden, 1996; Rubini and Meneghetti, 2001) are studied for signal processing. Typically, these meth-

ods transform a time-domain signal into another domain, with an attempt to extract the useful information embedded within the time series that is otherwise not readily understandable in its raw form. Although these techniques are widely applied in the waveform type time series, further signal processing methods are needed to extract information from multidimensional data.

In later signal processing studies, Kalman filter (Peng et al., 2012a), principal component analysis (Liao and Sun, 2011) and Gaussian kernel smoothing (Wang, 2010) are used to perform dimension reduction and data filtering. The main advantage of these methods is the ability to reduce noise in raw signals; however, when a common data source is available and different operational trajectories are included, the population of the entire dataset should be considered for signal processing. To that end, Wang et al. (2008) and Peel (2008) introduced the multi-regime data normalisation technique, which performs a standardisation method based on the population parameters of the entire dataset. Different operational trajectories can be filtered into a common range with respect to their initial wear levels and failure points. The main drawback in such dimension reduction methods is the introduction of novel trajectories that change the population characteristics and force the repetition of dataset standardisation.

Alternatively, the neural network based prognostic approaches are proposed (Heimes, 2008; Greitzer et al., 1999; Fink et al., 2014; Wu et al., 2016; Loutas et al., 2017; Elforjani, 2016; Yang et al., 2016; Zheng et al., 2017) and these could filter the data sets even though novel trajectories are introduced. The network function is able to standardise various trajectories without requiring population characteristics but these trajectories have distinct initial wear levels which should be considered with the entire dataset. Taking into account the associated merits and drawbacks, multi-regime data normalisation and neural network filtering can be merged to provide a signal processing

method that can produce desired outcomes with due consideration to dataset population parameters and wear-level characteristics.

2.1.3 Decision Making

After the data is collected and processed, the maintenance decision is made to provide sufficient and efficient support in terms of maintenance personnel's judgements on maintenance actions. This support within a CBM process can be further analysed by diagnostics and the prognostics (Jardine et al., 2006a).

Diagnostics deal with the detection, isolation and identification of the system state, while prognostics are concerned with the prediction of its future behaviour and the remaining life time of the system (Efthymiou et al., 2012). The main distinction between them is the nature of their analyses. Prognostic applications are based on prior event analysis, whilst a diagnostic application is concerned with posterior and current event analysis (Butler, 2012). Prognostic use of CBM can prevent faults or failures, and avoid further unplanned maintenance costs. These features help prognostic applications to gain advanced practice qualifications in maintenance but, as some faults and failures cannot be predicted in any way, prognostics cannot fully substitute for diagnostic use (Jardine et al., 2006a). Both of the applications can be practised in similar machinery domains and such successful implementations can be seen in the literature such as Elasha et al. (2014a,d, 2015a,c, 2016, 2014b).

2.2 Prognostics and Definitions

The ability to perform accurate and reliable prognostic estimations is a key concept in CBM management, and is additionally of critical value to improving safety, maintenance scheduling, mission planning and lowering costs and down time (Peng et al., 2010). Prognostics addresses the use of automated

procedures to detect, diagnose, and analyse system degradation, and to estimate RUL within the bounds of acceptable operating conditions before the occurrence of a failure or intolerable performance degradation levels. In order to provide sufficient lead time to maintenance personnel, the success of a CBM strategy is subject to these automated procedures of prognostics, which are responsible for sending out prior notices about the pending equipment failure (Butler, 2012).

In the development of a prognostic method, the main objective is to predict the failure time, at which a component or a system cannot complete its desired functions (Pecht, 2008). Predictions can be made by understanding the current system condition processes and the historical conditions that will affect the future behaviour of the system (Goebel et al., 2013). Thereupon, prognostics are always associated with other components of CBM.

Since a prediction is a statement about an uncertain event, the main prognostic approaches are concerned with basic assumptions regarding the characteristics of system degradation, and the science of prognostics is based on the following fundamental notions (Uckun et al., 2008):

- All systems deteriorate as a result of time, usage, and environmental conditions.
- Accumulation of damage and ageing are monotonic processes that disclose themselves in the physical and chemical composition of the systems.
- Ageing symptoms are detectable prior to system failure.
- Correlation of ageing symptoms is attainable with a model of system degradation and, therefore, the RUL of individual systems can be predicted.

Regarding these notions, Figure 2.2 represents a typical *initial fault to failure progression timeline* of a system (Hess et al., 2005). The timeline

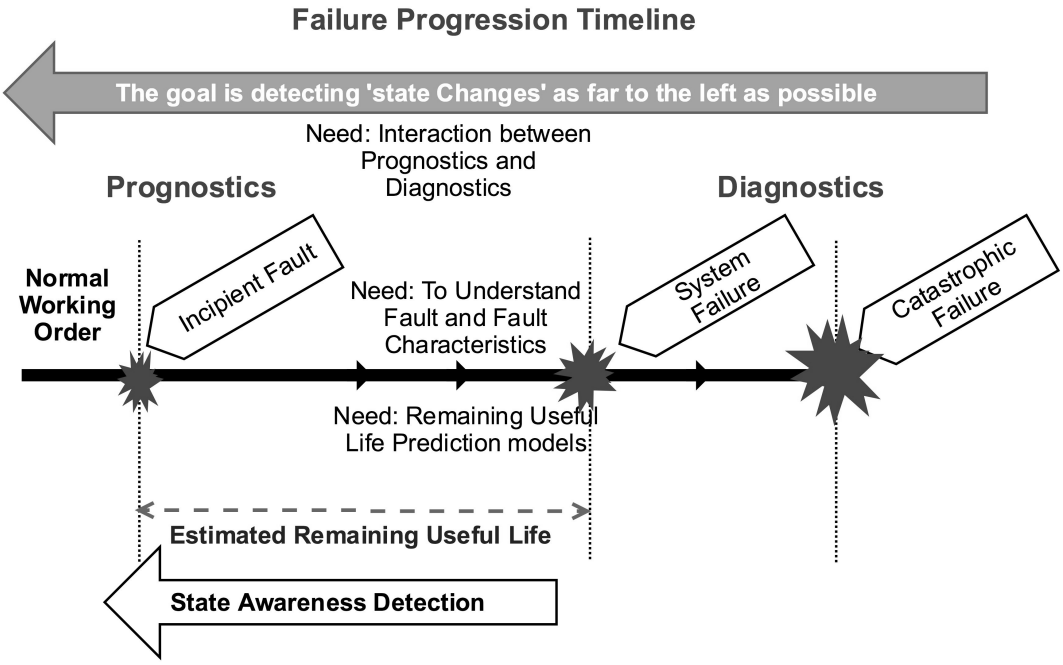

Figure 2.2: Failure progression timeline (adapted from Hess et al. (2005))

explains the role of prognostics and how the connection between prognostics and diagnostics can be achieved. At the beginning of a system's lifetime, all components are in proper working order and maintenance is not necessary. Each operational trajectory has its own specific initial health level, which is generally stable during the early stages of use. This continues until a critical period where an early incipient fault condition occurs. As time progresses and operation continues, the risk of system failure, which can cause a system damage and eventually a catastrophic failure, grows with time. Note that system failure and catastrophic failure are two different points in time. The initial detection of these failures and damage is always crucial to the estimation of RUL. The detection of state changes and fault characteristics requires interactions between prognostics and diagnostics. The goal in the timeline is improving state awareness detection as close as possible to the point of the first incipient fault occurrence.

In Figure 2.3 (Lewis, 2017; Goode et al., 2000), the system health continues to decrease due to the degradation process from an initial problem in the system, and eventually reaches a critical state that causes functional failure. The system starts with a certain level of initial health and manufacturing variation that can be considered normal, i.e., it is not representative of a fault condition.

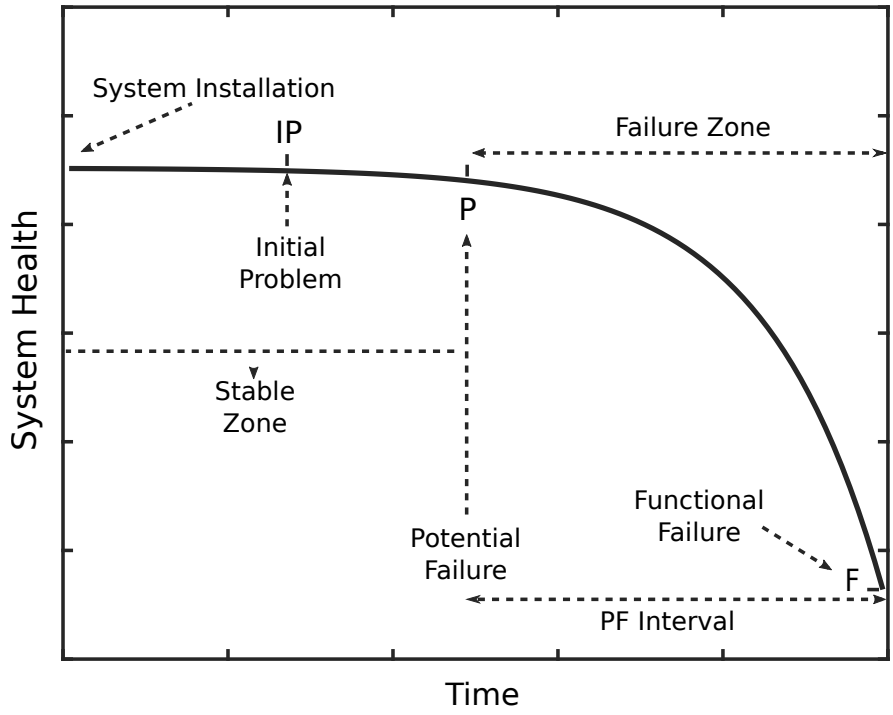

Figure 2.3: System life model (adapted from (Lewis, 2017; Goode et al., 2000))

The potential failure point defines the transition from stable zone to functional failure point. Stable zone is the time from initial machine installation to the potential failure point. This interval contains largely condition-monitoring measurements which are randomly varying around the higher system health limit. When the health index derived from condition monitoring readings exceed the alarm limit of potential failure, it is assumed that the system has entered the failure zone and will deteriorate, at an exponential rate, towards a functional failure Goode et al. (2000). Based on the location of these

zones and failure points, prognostics relate to the task of making multi-step ahead life prediction.

The prognostic prediction is practised between the initial detection of the failure and the progression to actual failure conditions (Lee et al., 2014). Since the lifetime estimation is applied within the CBM domain, it includes similar stages regarding data acquisition, signal processing and diagnostics. For example, a typical prognostic application is a sequential process with the following major stages in which the applications of various other CBM applications can be found (Tobon-Mejia et al., 2010; ISO-13381, 2015).

- **Pre-processing** describes any type of processing performed on raw data and pre-evaluation of existing failure modes. This step identifies the symptom relations regarding the performance and determines the potential impending failure modes.

- **Existing failure mode** is a step-by-step approach to identifying all existing failures in the processed data.

- **Future failure mode** estimates the most likely future modes, and their influence factors. Additionally, the estimated time to failure is calculated in this mode.

- **Post-action prognostics** proposes the maintenance actions that need to be done. To avoid the effects of undesired failure modes, the post-action prognostics are applied with consideration to the recommended actions.

Figure 2.4 demonstrates the above-mentioned basic stages of prognosis. The pre-processing phase receives the machinery information from sensors and identifies important failure features that are useful for determination of any fault conditions. Then, in the existing failure mode, failure mode analysis is

applied for feature extraction and fault classification. Considering the current fault level in the system, multi-step prediction is performed to calculate the fault evolution in the future failure mode. After receiving an estimated time-to-failure, the maintenance scheduling is performed to keep the prognostic application as reliable as possible.

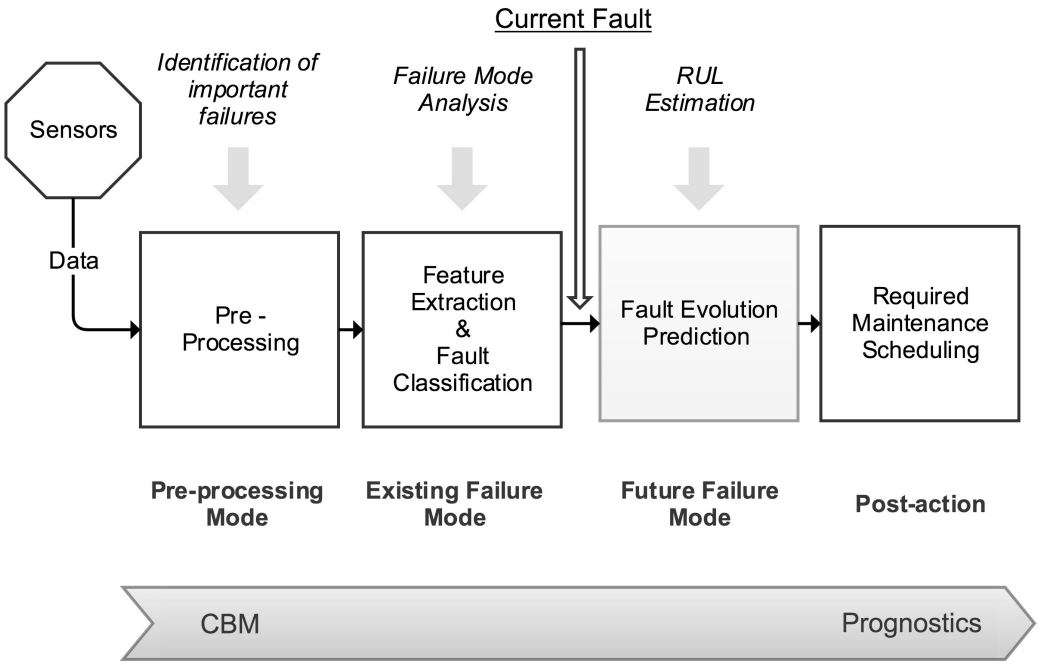

Figure 2.4: A Typical Prognostic Application (adapted from (Vachtsevanos et al.))

2.2.1 Prognostic Definitions

Common terms used in prognostic practice and their definitions have been reported in several studies (Wang et al., 2004; Muller et al., 2008; Lebold and Thurston, 2001; Byington et al., 2002). Apart from the terminology differences, these definitions mostly agree regarding the prediction aspect and RUL estimation of the system failure (Tobon-Mejia et al., 2012). Some of these prognostic terms are demonstrated in Figure 2.5 (Jones et al., 2001), where

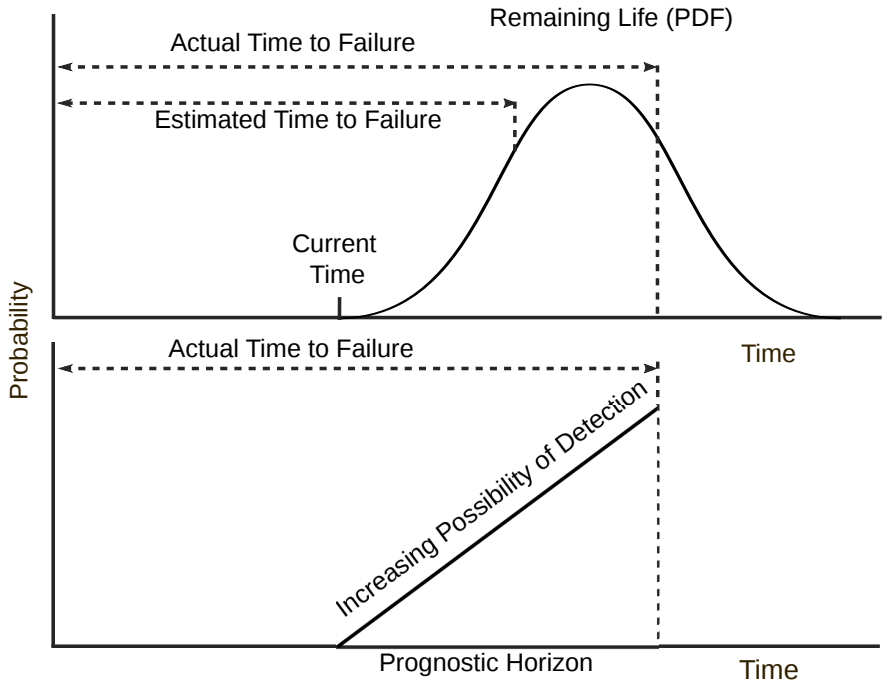

Figure 2.5: Prognostic Terms

the definitions are presented for a continuous time. The terms given here are used interchangeably with similar expressions by different researchers. It is worth noting that this section aims to reduce the uncertainties through giving the following terminology rather than introducing a new set of prognostic standards.

Actual time-to-failure (ATTF), or True Remaining Useful Life, is an unknown variable of prognostics that can only be known after a failure occurs in the system (Jones et al., 2001). The deviation between this variable and the estimation is of critical importance to prognostic prediction accuracy.

Estimated time-to-failure (ETTF) is the amount of time from the current moment to the point when the system is predicted to fail in its functions (Medjaher et al., 2013). Industry standard ISO-13381 (2015) defines prognostics as ETTF and risk of failure modes. Accordingly, in the prognostic literature, ETTF is generally the principal centre of studies, and the main

approaches to it are mostly concerned with the estimation of RUL (Efthymiou et al., 2012).

RUL, as a random variable based on the current age of the system, is expressed by (Jardine et al., 2006a):

$$X - t|X > t, Z(t), \tag{2.12}$$

where X is the variable of RUL and Z is the preceding condition profile until current age (t). RUL is stated in the time units used in the principal measurement of the overall system. For example, the measurement correlating with operations in commercial aircraft is cycles, hours of operation in jet engines, kilometres or miles in automobiles or number of pages in printers (Uckun et al., 2008).

The conditional variable of RUL corresponds to ETTF, and it is regarded as a basic prognostic measure representing the expected time until the occurrence of the first failure of a system.

$$E\left[X - t|X > t, Z(t)\right] \tag{2.13}$$

A key concept in the RUL framework is the failure probability density function (PDF) (Roemer et al., 2006) which defines the relative likelihood for the random variable of RUL in prognostic applications.

PDF, in a prognostic estimation, defines the density of a continuous time variable whose value at any given point in the sample index can be interpreted as providing a relative likelihood for the RUL. Defining X_t as the RUL variable at time t, the PDF of life estimation is conditional on observed condition monitoring or the history of operational profiles, θ (Si et al., 2011).

$$f\left(X_t|\theta_t\right) = f\left(X_t\right) = \frac{f\left(t + X_t\right)}{R_t} \tag{2.14}$$

where R_t is the survival function at time t and $f(t + X_t)$ is the probability distribution of remaining life at $t + X_t$. The calculated PDF can be used to specify the probability of the RUL falling within a particular range of values, as opposed to taking on any single ETTF value.

Prognostic Horizon (PH) is a measurement that calculates the foresight capability limits and the accuracy of a prediction algorithm for prognostic applications. PH can be formulated as the difference between the time index "t" when the predictions are first performed to meet the intended performance criteria and the time index for "end of prediction" (Saxena et al., 2009a).

$$PH = EOP - t \tag{2.15}$$

2.3 Review of RUL Prediction Methods

Originating from the initial concept of RUL prediction, a wide variety of prognostic techniques with many tools and methodologies have been described in the literature, and they continue to evolve in effectiveness of their predictions in different domains (Lee et al., 2014). In general, the current prognostic methods can be categorised into three main classes according to the way that they participate in prediction and forecasting: physics-based, knowledge-based and data-driven approaches (Peng et al., 2010; Zaidan, 2014; Eker, 2015).

2.3.1 Physics-Based Models

Physics-based models (PbM) consider the physical processes and interactions between the equipments and the failure mechanism in a system (Eker et al., 2014). The main assumption of this definition is that a physical model can define the evolution of degradation, and for this reason, PbM is often stated as a degradation model (Kim et al., 2017). Comprehensive mathematical models,

such as differential equations, are used to represent the degradation of system performance and physics-of-failure (Zaidan, 2014). These models require specific knowledge and theories relevant to a particular monitored system.

Table 2.2: Physics Based Models

Merits	Limitations
Paris & Erdogan Law Models (Li et al., 1999, 2000): Uses Paris' Law (Paris and Erdogan, 1963) for crack growth modelling.	
Least-square scheme provides adaptation of model parameters to condition changes.	Assumes that defect size is linearly correlated to vibration. Constants of materials are determined empirically
Forman Law (Oppenheimer and Loparo, 2002): Crack growth is modelled by using the Forman law of linear elastic fracture mechanics	
It can relate monitoring data with crack growth	Simplifying assumptions in the model
Paris crack modelling with FEA (Li and Choi, 2002; Li and Lee, 2005): Finite Element Analysis is used with Paris & Erdogan Law to calculate stress and strain fields	
It enables stress calculation	Performance is based on the accuracy of estimated crack size
Yu-Harris Models (Orsagh et al., 2003, 2004; Kacprzynski et al., 2004) :Uses Yu-Harris bearing life equation to predict spall initiation	
Based on cumulative damage with consideration to operating conditions	Uncertainty in different factors limits reliability
Contact analysis (Marble and Morton, 2006): Uses Finite Element Analysis for material stress field calculation	
It uses principles of damage mechanics	Different physics parameters are needed

Table 2.2 contains various important methods in the field of PbM prognostics. The presented mathematical models of degradation are typically employed in applications that are tied to health levels. PbMs are designed to have a combination of fault growth formulas and knowledge of the principles of damage mechanics. These models assume that an accurate mathematical model for component degradation can provide adequate knowledge for prog-

nostic outputs.

Crack growth modelling has been a common approach in PbM. The Paris and Erdogan Law (Paris and Erdogan, 1963) is used in several applications to relate the stress intensity factor range to crack growth under a fatigue stress regime.

$$\frac{da}{dN} = C\Delta K^m \tag{2.16}$$

where a a is the crack length, C and m are constants (that depend on the material characteristics, environment and stress ratio), $\frac{da}{dN}$ is the crack growth rate, and ΔK is the range of stress intensity factor during the fatigue cycle.

Li et al. (1999, 2000) modelled defect growth rate of rolling element bearings using a variation of Paris' Law, which states that the rate of defect growth is correlated to the effected area. The predicted defect size is compared to the actual defect size, and a recursive least-square scheme is applied to provide an adaptive prognostic model to the difference in defect growths. As presented in their numerical simulation tests, a slight parameter difference might cause large prediction errors. Similarly, Li and Choi (2002) and Li and Lee (2005) introduced a Paris' Law crack growth modelling with Finite Element Analysis (FEA), which enables material stress estimation based on defect size, bearing geometry, load and speed. The performance of their method relies on the accuracy of crack size calculation based on vibration data, and the calculations performed by the model are computationally-intensive in order to evaluate the probability of an observation.

Another growth model used in PbM is the Forman law of linear elastic fracture. Oppenheimer and Loparo (2002) related condition monitoring data and Forman law crack growth physics to life models. Since the instantaneous defect area size is identified during operations, the application might be impractical in some settings; thus, this crack growth modelling is based on

the direct estimation of defect area size from vibration data. In their study, the simplification of certain assumptions may need to be examined and the model parameters must be determined before application to specific machinery (Jardine et al., 2006a).

Orsagh et al. (2003, 2004) employed a stochastic version of the Yu-Harris life equation for fatigue spall initiation and the Kotzalas-Harris model to estimate failure progression and time-to-failure. After sensed data aggregation has taken place, current bearing health is estimated by calculating the time-to-spall initiation, after which the future bearing health module is predicted. Therefore, their fatigue model parameters were received from data using both physics-based and empirical models. Kacprzynski et al. (2004) improved this model by integrating material-level models, system-level data by merging algorithms and parameter tuning techniques, and proposing a framework for PbM for the spiral bevel pinion gear of a helicopter gearbox. The uncertainty in model factors such as gear geometry and the properties of contact, load and material might limit the reliability of prognostic applications. Different data forms, such as variation in fatigue and fracture or compressive stresses, have a particular influence on prediction accuracy.

An alternative PbM in spall progression was introduced by Marble and Morton (2006). They described a comprehensive experimental study of bearings by using FEA to estimate spall size, material stress surrounding the spall, rolling element load and speed. The model predicts the remaining cycles-to-failure with consideration to damage mechanics principles.

Although there are multiple application domains and the presented models differs from each other, the above-mentioned models have common features that make them suitable for specific purposes. In general, PbM approaches are relatively conventional and use pre-defined mathematical methods to understand component failure mode progression (George et al., 2006). They

show high accuracy compared to data-driven approaches in cases where a representative mathematical model is available; however, this type of approach may not be effective for RUL estimation in complex systems because the fault prediction in PbM provides a characteristic model for specific components rather than complex systems as a whole (Heng et al., 2009).

PbMs also suffers from the drawback that models are computationally expensive to develop, because the specific domain experts are required to be involved and the parameters in model need be validated by large sets of real-life data (Zaidan, 2014). Since PbMs are component specific and cannot be applied to other domains without re-assessing the parameters, most prognostic problems that can be solved by the physics-based approaches are at component level or subsystem level (Brotherton et al., 2000). For prognostics at system level (particularly for a complex system working under multiple operational conditions and producing multidimensional and multivariate signals from its sensors), it is difficult to construct an efficient PbM strategy that can mimic the dynamics of system's long-term degradation (Zaidan, 2014). Besides, it is almost impossible to describe the behaviour of each component of a complex system with a unique mathematical equation. The identification of failure modes under different operating conditions and model parameter recognition requires extensive experimentation which is often built on a case-by-case basis (Liao and Köttig, 2014). Hence, a certain PbM method designed for a specific system is not generally applicable to a different system without a significant amount of effort.

2.3.2 Knowledge-based Models

In the absence of an accurate mathematical model for a complex system with prior principles, the knowledge-based models (KbM) requiring no physical model appear to be promising in real-world applications (Peng et al., 2010).

Knowledge (or experience) based prognostics are simple and easy to perform applications where the historical failure information of systems are available for RUL prediction (George et al., 2006). They evaluate the similarity between an observed phenomena and a library of previously defined failures and deduce RUL expectancy from previous events (Sikorska et al., 2011). Table 2.3 shows the major applications of KbM in prognostics.

Table 2.3: Knowledge Based Models

Merits	Limitations
Expert Systems (Butler, 1996; Biagetti and Sciubba, 2004): algorithms mimicking the human expert to solve problems	
Effective on particular domains where there is sufficient information available	Unfavourable in novel cases where there in not sufficient knowledge
Fuzzy Logics (Feng et al., 1998; Satish and Sarma, 2005; Dmitry and Dmitry, 2004): a form of multivalued logic system	
Applicable for imprecise data	Needs rule definition for complex cases
Similarity-based Predictions (Wang et al., 2008; Wang, 2010; Eker et al., 2014; Lam et al., 2014; Ramasso, 2014a,b; Bektas et al., 2017): Based on the pairwise distance evaluation defined on two degradation trajectories	
High prediction accuracy and ability to reduce prognostic risks	Requires multiple run-to-failure data to estimate RUL

A typical example of KbM is the expert systems (ES) which can be defined as an automated representation of a computer system that is programmed to exhibit how a human expert solves a particular domain problem (Liao and Köttig, 2014; Peng et al., 2010). They are based on the rules expressed in the form of '*IF-THEN*" statements acquired from collections of real experiments (Sikorska et al., 2011). These rules can be either heuristic rules or specific domain rules, and they can be chained together using logical operators (Garga et al., 2001). As an example, Butler (1996) introduced an ES for incipient failure detection and predictive maintenance system to assess the integrity

of a power distribution system component and predict the maintenance requirements. Their system includes an expert system engine, a knowledge base model, mathematical and network models of equipment ageing and historical readings. Biagetti and Sciubba (2004) also designed an ES based prognostic and intelligent monitoring expert system that can produce real-time information about the existence of faults, estimate future time for detected and likely faults, and provide suggestions on problem controlling. However, there is no RUL information provided with their method.

Although the outputs of ES methods are comprehensible and reasoning can be evaluated with respect to a particular result, it is not always practical to acquire domain knowledge and convert it to rules, especially when novel situations are not covered explicitly or the system complexity increases (Zaidan, 2014)

Similar to ES, fuzzy logic (FL) is a problem-solving model that provides a simple way to arrive at a definite conclusion based upon imprecise inputs (Peng et al., 2010). A fuzzy system consists of a knowledge base variables to provide intuitive and human-like representation and reasoning with incomplete information. One of the earliest applications of FL in KbM prognostics was used in a chemical pulp mill for incident prevention and real-time process condition monitoring (Feng et al., 1998), and extended into prognostics of bearing faults in induction motors (Satish and Sarma, 2005). Dmitry and Dmitry (2004) also developed such a FL process. Their model maps inputs into fuzzy variables (fuzzification) and uses functions to de-map these variables into numerically precise outputs (defuzzification). Majidian and Saidi (2007) conducted a comparison of FL with a data-driven prognostic model for RUL prediction of boiler tubes and suggested that data-driven model performed better where its applicability is favourable compared to FL model.

FL has become an effective KbM for certain cases in which there is

inaccurate input information and/or no mathematical model is available to implement; however, it greatly depends on the availability of an expert to define the rules underlying system behaviour and design the fuzzy sets to present each variables characteristics (Zaidan, 2014; Sikorska et al., 2011). The challenges of FL implementation becomes more apparent in the case that experts require to define the rules for a complex system which potentially includes numerous interrelating components.

An alternative KbM is similarity-based prognostics. Despite this approach is categorised under data-driven models by several studies (Liao and Köttig, 2014; Eker et al., 2014; Mosallam et al., 2016), it follows the characteristics of KbMs such as similarity evaluation between monitored cases and using the library of degradation patterns for RUL estimations. However, similarity-based prognostics do not model the experience of a domain expert.

Wang et al. (2008) introduced the use of a similarity-based approach for RUL estimation for the International Conference on Prognostics and Health management data challenge, and received the leading score in the competition. Their systematic method includes the calculation of a health indicator for each unit and using the library of degradation patterns from the training units. The approach is particularly suitable for cases where a sufficient number of complete run-to-failure operational data is accessible. Trajectories from multiple units of the same system are used in a collaborative manner to create a library of degradation patterns. In order to predict the remaining lifetimes of the test units, the filtered degradation patterns for the test trajectories are matched to the pre-filtered training patterns in the library. The pairwise distance between pairs of training and test patterns is used to find the best matching position. The actual remaining life of those matched units is then used as the basis for estimation. The distance can be expressed as the average Euclidean distance over multiple trajectories between full degradation models

(training trajectories, q) and a test instance (test trajectory,p).

$$distance_i = \sqrt{\sum_{i=1}^{l}(q_i - p_i)^2} \tag{2.17}$$

Use of similarity based prognostics was extended by Wang (2010); Eker et al. (2014); Lam et al. (2014); Ramasso (2014a,b); Bektas et al. (2017) into RUL prediction of complex systems. The proposed algorithms are shown to be very effective in performing RUL prediction but they require a systematised data-processing method. Therefore, the above-mentioned similarity-based algorithms are used along with data-driven approaches for data processing.

2.3.3 Data-driven Models

Data-driven models are designed by processing monitoring data rather than by building mathematical models on broad system behaviours or using predefined system formulas (Schwabacher and Goebel, 2007). A typical data-driven model involves the determination of precursors to failure and remaining time by taking into account past records and estimation outputs from monitoring data (George et al., 2006). Data-driven approaches are calculated by regularly analysed periodic condition-monitoring data from system indicators (Peng et al., 2010). Such prognostic models, as trained by stored data, have been found to be more effective in numerous practical cases because of their simplicity in data discovery and consistency in complex processes (Heng et al., 2009). They can be carried out by automatic algorithm processing of historical degradation patterns in order to estimate future degradation (Pecht, 2008). These features of data-driven models allow the integration of innovative and conventional approaches to generate an inclusive prognostic method over wide-ranging data series (Byington et al., 2004a). As a result, the physical modelling of a system is no longer necessary, the algorithms' run-times become faster and the

collaboration between operations would become more practical.

Data-driven prognostics are based upon learning techniques which come from the theory of machine learning and pattern recognition. They range from conventional numerical methods of stochastic and statistical approaches, to black-box methods based on advanced machine learning techniques such as the neural networks (Sikorska et al., 2011). An ideal data-driven framework uses one or more of these learning techniques to perform fault detection (recognising an inadequate feature), fault isolation (identifying the fault location), fault identification (determining inadequate feature and the fault mode), and fault prognostics (estimating when a failure will happen) (Schwabacher and Goebel, 2007). Table 2.4 lists the existing major data-driven prognostic models and summarises their merits and limitations.

Table 2.4: Data-driven Prognostics

Merits	Limitations
Stochastic Models	
Bayesian Networks (Jensen, 1996; Zhang et al., 2007) : Also called as Belief Networks. A probabilistic model that represents a set of random variables and their conditional interdependency relations	
Requires less parameters to calculate	Lack of accuracy in complex systems
Particle Filter (Orchard and Vachtsevanos, 2009; Orchard et al., 2005; Saha and Goebel, 2009; An et al., 2013; Miao et al., 2013; Wang and Gao, 2014): A set of Sequential Monte Carlo (SMC) methods to solve filtering problems in data processing and Bayesian statistical inference	
Ability to combine reliability and monitoring information	Poor accuracy performance with high-dimensional data
Kalman Filter (Hu et al., 2012a; Julier and Uhlmann, 1997; Swanson, 2001): Also known as linear quadratic estimation (LQE). Use conditional information and produces estimates of unknown variables	
Tend to give more precise estimations than those based on a single measurement alone	Limited to linear cases. Poor accuracy performance with high-dimensional data

Hidden Markov (Bunks et al., 2000; Camci, 2005; Baruah and Chinnam, 2005; Dong and He, 2007; Ramasso, 2009) : It is modelled as a Markov process with unobserved (hidden) state recognition in various modes	
Simplicity in design	Require large amount of data for accurate modelling
Statistical Models	
Trend Extrapolation : (Batko, 1984; Kazmierczak, 1983; Cempel, 1987)	
Multi-step-ahead time series prediction	Poor performance when the degradation process is not mature enough
ARMA(Yan et al., 2004; Galati et al., 2008): Auto-Regressive Moving-Average	
Require small amount of historical data	Short-term prognostic horizon
Multi-regime Normalisation	
Reduce high dimensional data	Available only for data processing
Principal Component Analysis (Guo et al., 2002; Wang, 2010): A dimensionality reduction model by transforming the original features	
Reduces data sets to lower dimensions	Lack of population standardisation as a whole
Multiple Linear Regression (Ramasso, 2014a,b; Juesas et al., 2016; Bektas et al., 2017)	
Applicable on multi-dimensional data	Difficulties in specifying individual trajectory characteristics
Artificial Neural Networks	
Artificial Neural Networks (Heimes, 2008; Li, 2002; Bektas and Jones, 2016; Bektas, 2015; Rigamonti et al., 2017): Simulation of biological neural network functions. It learns the complex relationships between input and output data	
Ability to work on filtering, fitting, clustering, classification and prediction applications	Lack of standard methods. Low performance on multi-step ahead predictions of exponential time series

2.3.3.1 Stochastic Algorithms

Data-driven prognostics based on stochastic approaches provide a probability distribution of RUL that may be analysed statistically but may not be esti-

mated precisely. These methods are, in general, Bayesian-based approaches, which predict the state of a process by a minimum prediction covariance obtained from measurements. They can estimate both current and future states of nonlinear systems, and they can predict RUL by tracking the trends of growing deterioration before the asset hits a prearranged threshold (An et al., 2013).

In Bayes' theorem, the probability of an event is described by prior knowledge of conditions related with that event. This relation forms a reference point for updating estimations with due consideration of relevant evidence (Bayes et al., 1763).

$$\mathrm{P}(A|B) = \frac{\mathrm{P}(B|A)\mathrm{P}(A)}{\mathrm{P}(B)} \tag{2.18}$$

where A and B are two different observable events and $P(B) \neq 0$.

- $P(A)$ and $P(B)$ are the probabilities of A and B respectively.
- $P(A|B)$, conditional probability, is the probability of A given that condition B is true.
- $P(B|A)$ is the probability of B given that condition A is true.

When there is useful condition monitoring data available, a Bayesian network can model the degradation changes over time and provide results for prognostics which are invariably undertaken using time series forecasting. The most common variants of such Bayesian networks used in prognostics include Particle filters (Orchard and Vachtsevanos, 2009; Orchard et al., 2005; Saha and Goebel, 2009; An et al., 2013; Miao et al., 2013; Wang and Gao, 2014), Kalman filters (Hu et al., 2012a; Julier and Uhlmann, 1997; Swanson, 2001)

and Markov models (Bunks et al., 2000; Camci, 2005; Baruah and Chinnam, 2005; Dong and He, 2007; Ramasso, 2009)

Particle filtering (PF) is known to be a powerful tracking technique based on sequential Monte-Carlo methodology for sequential signal processing. Orchard and Vachtsevanos (2009) pioneered this approach in prognostic estimations, and carried it out by the approximation of the conditional state probability distribution using a swarm of points called "particles". It is stated that the model can provide results in long-term predictions, and it is suitable for on-line implementation.

In PF, the Bayesian theorem is used in a sequential way with samples (particles) which have probability information of unknown parameters. This process is mainly based on a state transition function f_s and a measurement function f_m.

$$\ddot{x}_t = f_s(\ddot{x}_{t-1}, \ddot{p}_t, \dot{\varepsilon}_t) \tag{2.19}$$

$$\ddot{z}_t = f_m(\ddot{x}_t, \ddot{\varepsilon}_t) \tag{2.20}$$

where $\ddot{x}$ is the damage state, $\ddot{p}$ is a vector of model parameters, $\ddot{z}$ is measurement data, $\dot{\varepsilon}$ and $\ddot{\varepsilon}$ are respectively process and measurement noises, and t is time (Orchard et al., 2005). The filtering problem includes the estimation of $\ddot{x}_t$, given all the measurements up to time "t" ($\ddot{z}_{(1:t)}$). Considering the Bayesian theorem, this estimation can be formalised as the calculation of the distribution $\mathrm{P}(\ddot{x}_t|\ddot{z}_{(1:t)})$, which can be done recursively in estimation and update stages where the computations are carried out by Monte Carlo sampling.

In prognostics, the state transition function is referred to a damage model (An et al., 2013). If this model is accurately defined to represent the

governing system dynamics, PF can be applied into nonlinear systems. However, in a linear system, the Kalman filtering (KF) is optimal since they have lower computational requirements than particle filters. In the case of KF, the state and measurement equations reduce to the following forms (Orhan, 2012).

$$\ddot{x}_t = \boldsymbol{Fs}\ddot{x}_{t-1}, \ddot{p}_t, \dot{\varepsilon}_t \tag{2.21}$$

$$\ddot{z}_t = \boldsymbol{Fm}\ddot{x}_t, \ddot{\varepsilon}_t \tag{2.22}$$

where $\dot{\varepsilon}$ and $\ddot{\varepsilon}$ are Gaussian noises, and $\boldsymbol{Fs}$ and $\boldsymbol{Fm}$ are respectively the state evolution and measurement matrices which are assumed to be known. With these equations, KF estimates the state of a process and minimises estimation covariance by including the measurement related to the state. This filtering model can correct the estimations with the latest measurements to minimise state error covariance.

Since both PF and KF methods are based on time series and their relationship with prior conditions, their application is limited in complex systems operating under various conditions. A damage state at any given time instant may not match with the upcoming conditions. Additionally, it is difficult to describe the behaviour of damage model for multiple sensor data.

A simpler Bayesian network, Hidden Markov Model (HMM), is also applied in the field of prognostics by Bunks et al. (2000), Camci (2005), and Baruah and Chinnam (2005). Accordingly, Zhang et al. (2005) investigated the use HMMs in bearing fault prognosis and applied to obtain the degradation index of bearings. Dong and He (2007) and Ramasso (2009) extended the use of HMMs in prognostics as an estimation tool for representing probability distributions over sequences of observations. In these applications, it is assumed that the observation at "t" is generated by some process whose state $\mathrm{P}(\ddot{x})$ is hidden from the observer and the state of this hidden process

satisfies the Markov property (in which future behaviour is predicted from the current or present behaviour) (Ghahramani, 2001). These applications can be used to recognise different fault types and states, but they might have practicability issues in physically observing a defect in an operating unit (Heng et al., 2009). Similar to the PF and KF method, HMM also has difficulties with multi-dimensional data and requires accurate degradation modelling.

2.3.3.2 Statistical Algorithms

Statistical data-driven models estimate the damage progression based on condition monitoring information on similar machines. Their main difference from stochastic approaches is that they commonly provide precise estimations rather than a probability distribution. They are often used as an alternative or supplement to artificial neural networks when a suitable model is available to account for the dynamics of a system (Sikorska et al., 2011).

One of the earliest time series prediction and forecasting prognostics is the trend extrapolation methods (Batko, 1984; Kazmierczak, 1983; Cempel, 1987). The degradation pattern is associated with a single time series (either monitored or calculated) which is assumed to follow a monotonic trend. This single degradation parameter is plotted as a function of time and a threshold level is pre-established to decide the end of life point. The major advantage of such forecasting methods is the simplicity in their calculations which can be carried out on a basic programmable algorithm. However, their main drawback is that operating conditions are stable and/or do not have any affect on monitored time series.

In the statistical analysis of time series, autoregressive moving average (ARMA) models are widely used for modelling and forecasting time series (Box et al., 2015). The main idea of ARMA is to fit the data to a parametric time series model and extract features based on this model. It consists of

two parts: the auto-regressive part and the moving average part. The damage progression is obtained with curve fitting in moving average part, and added to the autoregressive model output to predict the future values (Liao and Köttig, 2014). For a time series $x = 1 : t$, the ARMA model with autoregressive model of order P and the moving average model of order Q takes the following form.

$$\ddot{x}_t = c + \varepsilon_t + \sum_{i=1}^{P} \phi_i \ddot{x}_{t-i} + \sum_{i=1}^{Q} \Theta_i \varepsilon_{t-i}, \tag{2.23}$$

where ϕ and Θ are respectively auto-regressive and moving average terms (Whitle, 1951). This representation is effective for short-term predictions, however, cannot provide reliable long term predictions due to the model's sensitivity to initial system conditions and systematic errors in the predictor (Box et al., 2015). Although, this problem can be minimised by avoiding past estimations for future predictions and being reliant on condition monitoring data (Wu et al., 2007), ARMA models are limited for non-stationary and dynamic processes.

The review of data-driven models in this section shows that due to the incomplete understanding on the multi-dimensional failure mechanisms, time series prediction and forecasting methods lack the ability to deal effectively with complicated multidimensional and noisy data. Further data processing methods are needed to deal with this efficiently.

The raw values of multi-dimensional time series, which are inconsistent with each other and operates under various conditions, need a feature extraction transformation of the multi-regime data in the high-dimensional space to a space of single health level dimension (Bektas et al., 2017). This transformation can reduce the dimensionality of the time series from their original scales to a notionally common scale that will include meaningful information for prognosis.

Such feature extraction and dimension reduction can be combined in one

step by using regression models which perform a mapping of the multi-regime data to a lower-dimensional space in such a way that the variance of the measurements in the low-dimensional finding is maximised. Ramasso (2014a,b) applied a multiple linear regression model for complex systems operating under different conditions. Their model could standardise the multi-regime data into a common space for further time series prediction and forecasting prognostics. In similar complex system domains, Juesas et al. (2016); Bektas et al. (2017) extended the approach and applied into alternative estimation models.

Multiple linear regression calculates the relationship between different explanatory variables and a target variable by fitting a linear equation to observed data (Chatterjee and Hadi, 1986; Freedman, 2009). This model is based on:

$$y = x\beta + \epsilon \tag{2.24}$$

where y is a $n \times 1$ vector of values of the target variable, x is an $n \times p$ matrix of observed responses and β is a $n \times 1$ vector of coefficient estimates for a multiple linear regression of the responses.

$$y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_n x_n \tag{2.25}$$

More complex models may include multiple observations (multivariate time series) and the equation is modified by considering x as a matrix instead of a vector.

$$y_i = \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 x_{i,3} + \cdots + \beta_p x_{i,n}, \tag{2.26}$$

$$for \quad i = 1, 2, \cdots, p \tag{2.27}$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{bmatrix} \quad x = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix} \tag{2.28}$$

When datasets represent multiple instances and cover various realistic and difficult cases including different operating conditions and fault modes with unknown characteristics, the "regression" can tackle the problem of feature feature extraction and standardisation (Ramasso, 2014a). After the coefficients, β, are estimated for a specific case, they can be applied into similar matrix of observed responses. The major limitation in this model is that the coefficient estimation is a supervised method that requires pre-defined target variables. Although there are mathematical definitions are used to define these variables, the calculation of characteristic damage progression in individual complex systems and initial health level (or wear level) is a major issue to be considered.

Principal component analysis (PCA) is another statistical dimensionality reduction procedure that uses an orthogonal transformation to convert a set of correlated inputs possibly into a set of linearly uncorrelated principal components. In PCA, these components are obtained from singular value decomposition of rectangular matrices, x (Holland, 2008). To standardise these multi-dimensional data, the first principal components are used as the health indicator for prognosis (Ramasso, 2014a,b; Juesas et al., 2016; Bektas et al., 2017).

$$y_1 = \beta_{11}x_1 + \beta_{12}x_2 + \beta_{13}x_3 + \cdots + \beta_{1p}x_p = \beta_1^T x \tag{2.29}$$

Similar to the regression model, β is a matrix of coefficients that is determined by PCA. This dimensionality reduction method is not a super-

vised model and does not require a pre-defined target variables. However, the method can only be applied into individual run-to-failure trajectories. Considering a dataset with multiple component-wise cases, the damage progression and initial health level of individual cases cannot be standardised into a common scale.

In order to provide a common scale across all the characteristics of a dataset, normalisation is a common well-known pre-processing step to perform component-wise standardisation before the prognostic analysis. A standard method to achieve this is to use the standard score (Peel, 2008; Wang et al., 2008; Wang, 2010; Lam et al., 2014; Ramasso, 2014a,b; Malinowski et al., 2015; Rigamonti et al., 2016):

$$N(x^d) = \frac{x^d - \mu^d}{\sigma^d}, \quad \forall d \tag{2.30}$$

where x^d are the original data values (data set) for feature d (regime), and μ^d and σ^d are respectively the mean and standard deviation of the regime. Peel (2008); Wang et al. (2008) proposed such a component-wise "multi-regime normalisation" method to standardise the multi-regime sensor readings according to each other within the same domain. Unlike the regression analysis and PCA, their methodology can deal with the damage progression in complex systems and consider the population characteristics (μ, σ). They could standardise the entire dataset together with its all components and preserve the characteristic damage progression and initial health levels of multiple trajectories.

However, for the case considered in the works of Peel (2008); Wang et al. (2008), all trajectories were available at the same time. The "normalisation at once" has therefore not been a major issue. Nevertheless, it would be rather unlikely to find such data in a real-life scenario due to the restrictions on data proprietary considerations and confidentiality (Ramasso and Saxena, 2014).

In a real-world scenario, the "multi-regime normalisation" should be repeated for each novel incoming trajectory in order to calculate the changed population characteristics.

2.3.3.3 Artificial Neural Networks

One of the most commonly used data-driven approaches in the prognostic literature is the Artificial Neural Networks (ANNs) (Goebel et al., 2008b). ANNs are biologically inspired programs that are loosely analogous to the behaviour of the neural networks of the brain, and they are used as machine learning systems made up of data processing neurones which are the units of neural networks (Bishop, 1995; Kozlowski et al., 2001). The neurones establish a set of interconnected functional links between input series and a desired output where the connections can be calculated and trained for the optimal performance (Byington et al., 2004a). This connection is typically achieved by exposing the network to a set of input samples, training the network, and re-adapting the network to minimise errors (Bishop, 1995).

The computational model of ANN is a set of multiplication, summation and transfer functions (Krenker et al., 2011). The neurons practice the multiplication by weighting the inputs in the first step and then summing all weighted inputs. The sum of these weighted inputs is exposed to the transfer function. In this process, the weights used at the neurons are automatically changed to increase the compliance of the model with the data (Bishop, 1995). These connections and organisation of ANN are the key features in establishing a set of interconnected functional relationships between numerous input series and a desired outputs where the relationship can be trained for optimal performance (Byington et al., 2004a).

Neural networks are practical tools for effectively modelling engineering systems consisting of a broad category of non-linear dynamical systems, dimen-

sion reduction methods, regression analysis and discriminant models (Sarle, 1994). In certain complex engineering applications, the observations from a system may not include precise data, and the desired results may not have a direct link with the input data; in such cases, ANNs are a competent application to model the system without knowing the exact relationship between input and output data series (Murata et al., 1994). ANNs, therefore, are suitable for complex system prognostic algorithms and are faster and easier to calculate compared to various other prognostic methods. For all these reasons, ANNs have been widely employed as one of the most popular data-driven prognostic methods, and a significant number of studies across different disciplines have stated the merits of artificial neural networks through the introduction of numerous different methodologies.

Such an advanced neural network training is designed by Heimes (2008) to classify the difference between healthy and degraded condition monitoring time series of a complex system. Their network is designed with back-propagation through time gradient calculations and developed to solve the issues in adaptation, filtering and classification. Initially, a Multi-Layer Perceptron (MLP) neural network design is undertaken to determine whether the difference between a healthy system and a failed system can be classified effectively. MLP function predicts the number of cycles remaining before the failure. It is assumed that the earliest samples in each time series characterise a healthy time line, while the latest samples correspond to a degraded time line. The condition monitoring data is then clustered into two different parts: a healthy unit and a degraded unit. An MLP-based classifier could distinguish these units with 1% error rate. To handle the filtering within non-linear time domain dynamic system modelling, a recurrent neural network (RNN) structure with internal memory and feedback components is used to learn complex non-linear dynamic mappings. The RNN structure utilises all sensor data and

operating conditions as input series for estimation.

Peel (2008) also constructed a similar Multi-Layer Perceptron and Radial Basis Function networks as the regression models for prognostics. A Kalman filter method is combined with a proper selection of these networks. Their designed algorithm delivers a mechanism for fusing the Kalman filter and multiple neural network model predictions over time. Peel (2008) discussed that the data pre-processing and data exploration are essential initial stages of a successful prognostic framework for complex systems which operate under multiple operational conditions. They concluded that these stages result in the identification of different regimes from sensors.

Peng et al. (2012a) and Rigamonti et al. (2016) provided the echo state network-based prognostic model, which is an architecture and supervised learning principle for recurrent neural networks. This network model drives a random, large and fixed recurrent neural network via the input signal, thus inducing each neuron within the network to act as a non-linear response signal, and then combines a desired output through a trainable linear combination of all of the response signals. Abbas (2010) developed a further multilayer feed-forward neural network architecture using an error back propagation algorithm in which they employed to develop the creep damage predictive models for different regime phases, i.e., take-off, climb, and cruise. Wang (2010) also expanded the earlier trajectory similarity-based prediction method (Wang et al., 2008) with a Radial Basis Function Neural Network (RBFN) based RUL prediction method. It has been stated that the similarity-based model has shown considerable advantages in their predictive performance over ANN-based prediction methods.

In the field of complex systems with multi-dimensional condition monitoring data, Jianzhong et al. (2010) and Riad et al. (2010) extended the use of ANNs in an attempt to achieve better performance within the learning

methodology regarding RUL estimation of test trajectories. Javed et al. (2012) emphasised the neural networks in the feature selection procedure for prognostics with the intention of showing that feature selection should be performed according to the predictability of features. For multi-step ahead predictions, Bektas and Jones (2016) introduced a non-linear autoregressive neural network prognostic model as a form of dynamic filtering in which past values of a time series are used to predict future values. However, it was observed that this type of neural network has issues predicting the exponential behaviour of damage propagation. Therefore, a recurrence relation model was used to transform input and output data for network training.

In Table 2.5, ANN prognostic approaches applied in complex domains are described. Although the most attractive feature of these neural network applications is the accomplishment of their learning ability, it is not always possible to train the network as desired. The networks used at different phases of prognostics may not be as effective as expected, and this is generally more evident in time series showing complex degradation growth or decay. In such cases, the ANN structure behaves as an autonomous system which attempts to recursively imitate the dynamic system behaviour that caused the non-linear time series (Haykin and Li, 1995; Haykin and Principe, 1998). The multi-step ahead predictions of ANN applications can be quite challenging when only a few time series or a little previous knowledge about the degradation process is available and the failure point is expected to happen in the longer term (Menezes and Barreto, 2008).

ANNs, which are designed for one-step-ahead prediction models, include only actual sample points of the initial time series used in network modelling; the prediction tasks are modelled to estimate the next value of time series, without feeding externally back to the models' input values (Zemouri et al., 2010). In a longer multi-step-ahead prediction horizon, ANN model output

Table 2.5: ANN-based Prognostics in Complex Systems

Parker Jr et al. (1993)	A neural network pattern recognition model that uses a constrained, minimum-logistic-loss criterion for multi-class problems
Brotherton et al. (2000)	A classification model for complex systems that are difficult to model physically. The approach provides novelty detection capability in sensor data and thereby statistical state modelling of the complex system with respect to known faults.
Bonissone and Goebel (2002)	A systematic framework for building a model to estimate time-to-fail. Hybrid models of neural, fuzzy and evolutionary computation methods are applied to classification, prediction, and control problems.
Heimes (2008)	Applied to multidimensional dataset. MLP based classifier and RNN based structure for filtering within non-linear time domain of turbofan systems
Goebel et al. (2008b)	ANN is applied to learn the damage state of relatively sparse training sets with very high noise content (a rotating equipment in an aerospace setting)
Peel (2008)	The method involves the estimation of RUL of an unspecified complex system using a data-driven model combination of Multi-Layer Perceptron Radial Basis Function networks with a Kalman filter method
Baraldi et al. (2013)	A trained bagged ensemble of Artificial Neural Networks is embedded in the Particle Filtering method as an empirical measurement model

is required to be externally fed back to the initial time series for a fixed but finite number of steps; the regressing components of these input series, which are previously formed of actual sample points of the initial time series, are progressively replaced by already predicted values (Sorjamaa et al., 2007). Such a replacement might cause an imbalance in exponential curve predictions and the multi-step predictions can overly imitate the training data (Bektas and Jones, 2016).

However, in data filtering models for prognostics, ANNs provide a robust computational mapping between the raw data and a desired output to be used in network prediction (Demuth et al., 2008). Such neural network filtering

models have been applied in multidimensional data, and showed high performance in prognostic performance (Parker Jr et al., 1993; Brotherton et al., 2000; Heimes, 2008). Considering that the multi-step ahead predictions and dynamic modelling in ANNs are complex tasks, and play a challenging and significant role in ANN structure (Principe et al., 1999; Menezes and Barreto, 2008), a combination of ANNs with alternative methods is generally necessary in order to achieve better prognostic performance. Some of these various methods are shown in literature such as Parker Jr et al. (1993); Brotherton et al. (2000); Bonissone and Goebel (2002); Peel (2008); Baraldi et al. (2013); Bektas and Jones (2016). Each combination in these applications can be regarded as a hybrid model for prognostic frameworks.

Learning in a typical network is associative since the network learns an association between one type of data (input) and another (output) (Cross et al., 1995). When the network training is supervised by the target data, it knows the desired response and the actual outcome. Such supervised-type classification techniques tend to be more accurate since each classifier is trained by a representative data trajectory known as a corpus. In contrast, the unsupervised learning does not employ prior training to process the classification (Chaovalit and Zhou, 2005). In various prognostics models such as Ramasso (2009); Ramasso and Gouriveau (2010); Sarkar et al. (2011); Xue et al. (2011); Yu (2013); Lin et al. (2013); Tamilselvan and Wang (2013); Bektas and Jones (2016); Mosallam et al. (2016), the supervised classification stages is used for health indicator detection.

The generalised time-varying health index equation introduced by Saxena et al. (2008b) is commonly used as an additive term to yield supervised classifications for complex systems.

$$h(t) = 1 - d - exp(at^b) \tag{2.31}$$

where d is an arbitrary point in the wear-space, a and b are model parameters and t is time. This health index can also be used for different phenomena within a system. As an example, health can be described by the trajectories for flow (f) and efficiency (e) that might differ for various fault modes. Thereby, they are required to be as separate health-related indexes.

$$e(t) = 1 - d_e - exp(a_e(t)t^{b_e(t)}) \tag{2.32}$$

$$f(t) = 1 - d_f - exp(a_f(t)t^{b_f(t)}) \tag{2.33}$$

These terms are then aggregated to form the overall health index;

$$H_{(t)} = g(e_{(t)}, f_{(t)}) \tag{2.34}$$

where the function g corresponds to the minimum of all operative margins.

The main challenge in supervised prognostic approaches (as well as the neural network fitting approaches) is the identification of data characteristics. When a common dataset is available and used for predictions, a predefined model might show complications when attempting to define characteristics in different instances in the dataset. The most obvious examples of these complications are the initial wear levels and failure points. Since each operating trajectory would have a case-specific starting performance level and a mutual threshold level, the operational health level should be standardised by regarding these levels.

Alternative unsupervised learning algorithms for classification are used in the publications of Wang et al. (2008); Wang (2010); Peng et al. (2012a); Sarkar et al. (2011); Javed et al. (2013); Lam et al. (2014); Mosallam et al. (2016); Rigamonti et al. (2016) in order to perform dataset partitioning into

the operating regimes. These methods could be regarded as cluster analysis, which is used for exploratory regime analysis to identify hidden patterns in data trends. As these methods regard with the population parameters in the entire dataset, the characteristic trajectory features such as initial wear level and failure point could be standardised into a common scale. Since the supervised training stage are trained by the representative data trajectory, the identification of population parameters is disregarded. However, one can apply the unsupervised learning to take the population parameters into account and then use the outcomes in a more effective supervised model as an output data in a neural network filtering model.

2.3.3.3.1 Neural Network Architecture:

The structure of neural networks covers a broad area of study and it would be impractical to discuss all types of neural networks in this work (Hagan and Demuth, 1999). Instead, the common tools of neural network architecture for filtering and the multilayer perceptron will be concentrated on in the following sections. This architecture returns a common neural network fitting function with a hidden layer (Beale and Demuth, 1998). Similar neural network applications in prognostics can be found in the works of Greitzer et al. (1999); Fink et al. (2014); Wu et al. (2016); Loutas et al. (2017); Elforjani (2016); Yang et al. (2016); Zheng et al. (2017). Based on these works, this section defines a network model that uses the unsupervised HI outputs in the network training stage and generates a network function to fit the raw data trajectories.

The most essential building block in neural networks is the single-input neuron structure (Demuth et al., 2008), such as that shown in Figure 2.6. In this sample neuron, there are three definite functional operations, namely the weight function, the net input function and the transfer function (Demuth et al., 2008). The scalar input x is first multiplied by the scalar weight, w,

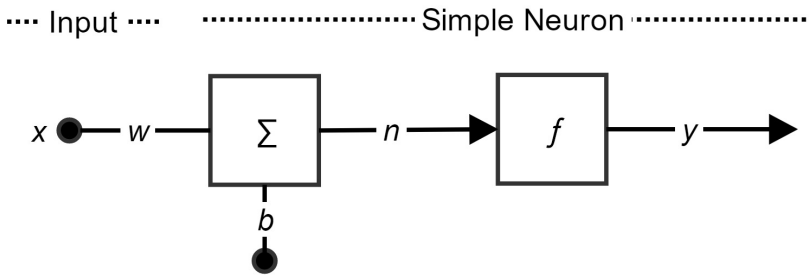

Figure 2.6: A single neuron

to form another scalar product, wx. The weighted input, wx, is added to the bias, b, to form the net input, n, in the second functional operation. The bias is such as a weight with the exception that it has a constant input of "1". In the final operation, n penetrates through the transfer function, f, which results in the output, y. The basic idea of a neuron is to adjust parameters x, w and b such that the network demonstrates the desired behaviour. Hence, the network can be trained to do a particular task by adjusting the weight and bias parameters.

For the neurone output, the general equation can be denoted by the following formula (Lippmann, 1988):

$$y = f(wx + b) \tag{2.35}$$

As ANNs are complex structures, a neuron typically has more than a single input. A network illustration with multiple inputs is demonstrated in Figure 2.7. All inputs, $x(1), x(2), x(3), \cdots, x(r)$, are weighted by their corresponding elements, $w(1,1,), \cdots, w(1,r)$ in the weight matrix, W (Hagan et al., 2002).

$$n = w_{1,1}x_1 + w_{1,2}x_2 + \cdots + w_{1,r}x_r + b \tag{2.36}$$

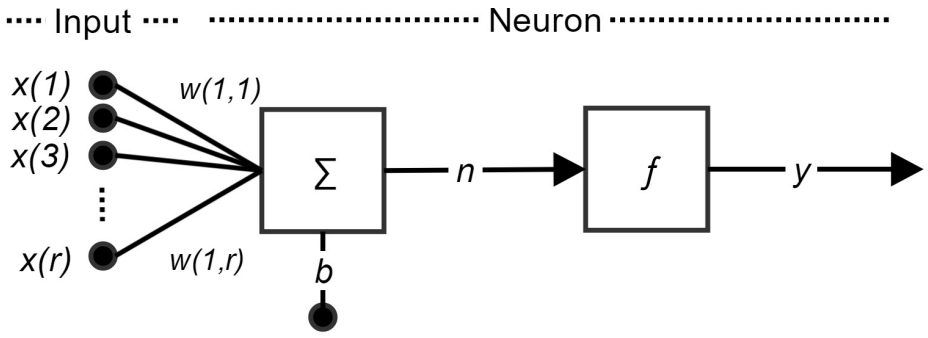

Figure 2.7: A single neurone with multiple inputs

This equation, in matrix form, can be expressed as:

$$n = \mathbf{W}x + b \tag{2.37}$$

where the weight matrix has a corresponding element for each input entering the network.

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,r} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,r} \\ \vdots & \vdots & & \vdots \\ w_{s,1} & w_{s,2} & \cdots & w_{s,r} \end{bmatrix} \tag{2.38}$$

and the scalar output of a neurone can be formulated as

$$y = f(\boldsymbol{W}x + b) \tag{2.39}$$

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2.40}$$

where the fixed real-valued weights, (w_i), are multiplied by state vari-

ables, (x_i), with the addition of bias, b. The neuron's output, y, is obtained as a result of the nodes and the transfer function of the neurons f (Barad et al., 2012; Krenker et al., 2011).

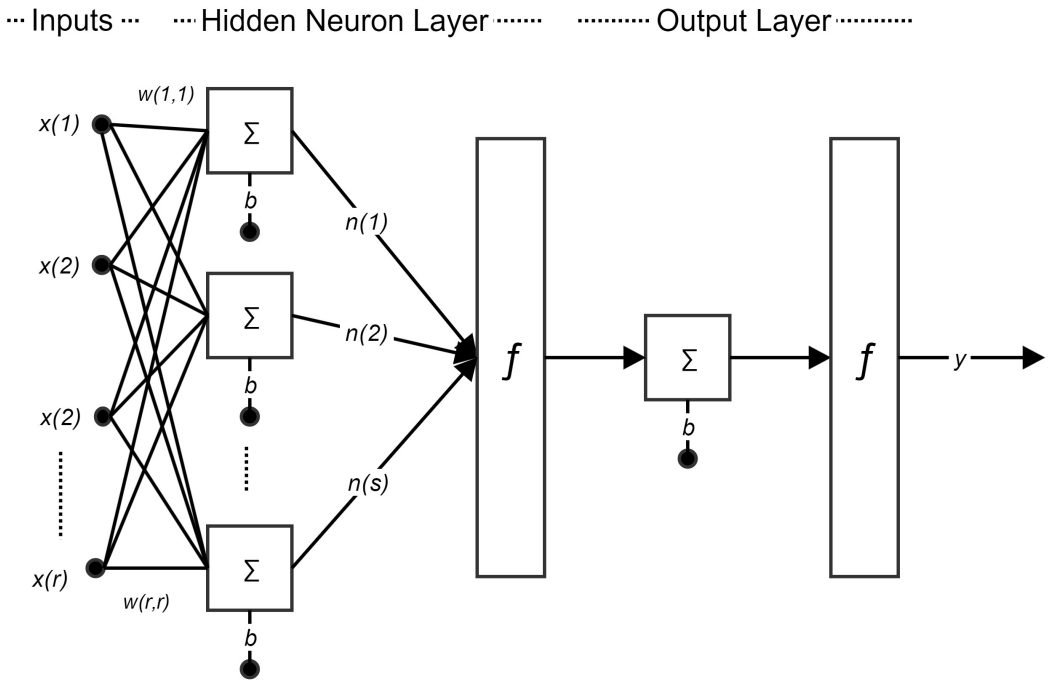

Figure 2.8: A multiple-layer neural network

Generally, one neuron, even if structured with many inputs, is not sufficient for training, and multiple neurones operating in parallel layers are instead necessary (Hagan and Demuth, 1999). In Figure 2.8, a multiple-layer neural network model is shown. Each layer here is formed of its own weight matrix, bias vector, transfer functions, and inputs and outputs. The demonstrated structure is formed of a feed-forward model that takes a set of input vectors (raw data) as columns in a matrix, and then arranges another set of output vectors (HI) into a second matrix.

Neurons, which are the building blocks of neural networks, evaluate these input state variables. This definition can be denoted by the following equation:

$$
\begin{aligned}
y_{(t)} &= f\left[\mathbf{x}_{(t)}\right] \\
&= f_o\left\{b + \sum_{h=1}^{n_h} w_h f_h \left(b_h + \sum_{i=1}^{n} w_{ih} x_{(i)}\right)\right\}
\end{aligned} \tag{2.41}
$$

where the network used for fitting function is a two-layer feed forward network with a sigmoid transfer function, f_h, in the hidden layer and a linear transfer function in the output layer, f_o (Demuth et al., 2008).

$$
f_h(n) = \tfrac{1}{1+e^{-n}}, \quad f_o(n) = n \tag{2.42}
$$

2.3.3.3.2 Neural Network Regularisation:

In the training of multilayer neural networks, overfitting and computational overheads might lead to poor network calculations, especially when the data is excessively complex and the parameters relating to the number of observations are outnumbered (Srivastava et al., 2014). In such cases, the network could memorise the samples for the training data, but it cannot properly generalise the upcoming testing cases (Lawrence et al., 1997).

One common network training approach to avoid overfitting is Bayesian regularisation (BR) method. Other well-known regularisation alternatives to this method are the "Levenberg-Marquardt" and "scaled conjugate gradient" practices (Demuth et al., 2015). In the case of the Levenberg-Marquardt model, the network training typically requires more memory but less time and it automatically terminates when generalisation stops improving, as defined by the increase in the mean squared error value of the validation samples. The main drawback of the Levenberg-Marquardt algorithm is that it requires the storage of some matrices that can be quite large for certain problems.

The "scaled conjugate gradient" needs less memory and can be used when the training data is short. Bayesian regularisation typically requires more time in comparison to other regularisation algorithms due to the adaptive weight minimisation, but it is able to provide satisfactory generalisations from difficult and/or noisy datasets (Demuth et al., 2008).

The function in the Bayesian regularisation method stands on the Gauss-Newton approximation to the Hessian matrix. The algorithms updates the network weights and biases in accordance with Levenberg-Marquardt optimisation. The regularisation reduces a compound of squared errors and weights in pursuant of diminishing the computational overhead, and then identifies the correct compound so as to provide a practical network generalisation. This definition of Bayesian regularised neural network is formed by the extensive works of MacKay (1992b) and Foresee and Hagan (1997). Based on their works, the detailed application of the Bayesian rule to neural network training structure and to optimising regularisation will be detailed in the following subsections.

2.3.3.3.3 Bayesian Regularisation Algorithm

In the network training process, the weights are calculated and adjusted with the objective of minimising the error function, E_D (Bui et al., 2012). The aim of this training is to produce a network that produces only small errors with the training data, but will also respond properly to any novel inputs that might be presented. When a network can perform as well on novel input data as on training inputs, it is accepted that the network generalises well.

To achieve the optimum generalisation, until the error function gives acceptable results, the errors are fed backwards through the neural network to adjust the weights. ANN uses this back-propagation training method to combine the sum of the least squares error function with a further term called

regularisation (Bui et al., 2012).

$$\begin{aligned} F_{(\mathbf{w})} &= (\beta E_D + \alpha E_{\mathbf{w}}) \\ E_{\mathbf{w}} &= \sum_{i=1}^{n_w} w_i^2 \end{aligned} \tag{2.43}$$

where $E_{\mathbf{w}}$ is the penalty term, and the terms α and β are objective function parameters, also called regularisation parameters. Their relative size is used to dictate the network learning emphasis.

- For $\alpha << \beta$, the learning algorithm continues to allow the errors to become smaller
- For $\alpha >> \beta$, learning interferes and emphasises weight size diminution at the expense of network errors, so that the network response becomes smoother, and the algorithm can generalise well.

The goal of implementing regularisation is that of defining accurate values for the objective function parameters (Foresee and Hagan, 1997). This is where the Bayesian method finds use.

To apply the Bayes' theorem in the neural network training process, the weight function, regarded as a prior distribution, can be turned into a posterior distribution (Bishop, 1995).

$$P(\mathbf{w}|D, \alpha, \beta, M) = \frac{P(D|\mathbf{w}, \beta, M)P(D|\mathbf{w}, \alpha, M)}{P(D|\alpha, \beta, M)} \tag{2.44}$$

Where;

- D corresponds to the data set
- M represents the neural network model used
- $\mathbf{w}$ is the vector of weights in the network
- $P(\mathbf{w}|D, \alpha, \beta, M)$ is the conditional probability

- $P(D|\mathbf{w}, \beta, M)$ is the likelihood function corresponding to the probability of error
- $P(D|\mathbf{w}, \alpha, M)$ is the prior density representing initial knowledge in the weights before the data is collected
- $P(D|\alpha, \beta, M)$ is the normalisation factor that ensures that the total probability equals 1 (MacKay, 1992b; Foresee and Hagan, 1997) .

Assuming that the data noise and the prior distribution generated for the weight are Gaussian, the above-mentioned probability densities can be expressed by the following equations (Foresee and Hagan, 1997; Burden and Winkler, 2009; Bui et al., 2012).

- **Prior** probability of weights

$$P(D|\mathbf{w}, \beta, M) = \frac{1}{Z_D(\beta)} exp(-\beta E_D) \tag{2.45}$$

- **Likelihood** as the probability of errors

$$P(\mathbf{w}|\alpha, M) = \frac{1}{Z_{\mathbf{w}}(\alpha)} exp(-\alpha E_{\mathbf{w}}) \tag{2.46}$$

where:

$$Z_D(\beta) = (\pi/\beta)^{n/2} \tag{2.47}$$

and:

$$Z_{\mathbf{w}} = (\pi/\alpha)^{N/2} \tag{2.48}$$

After these probabilities are substituted into equation 2.44, it becomes the following form:

$$P(\mathbf{w}|D,\alpha,\beta,M) = \frac{\frac{1}{Z_{\mathbf{w}}(\alpha)}\frac{1}{Z_D(\beta)}exp(-(\beta E_D + \alpha E_{\mathbf{w}}))}{P(D|\alpha,\beta,M)} \tag{2.49}$$

and since $P(D|\alpha,\beta,M)$ is the normalisation factor, the posterior probability becomes:

$$P(\mathbf{w}|D,\alpha,\beta,M) = \frac{1}{Z_F(\alpha,\beta)}exp(-F_{\mathbf{w}}) \tag{2.50}$$

The optimal weights are compelled to maximise the posterior probability and to minimise the regularised objective function.

$$F = (\beta E_D + \alpha E_{\mathbf{w}}) \tag{2.51}$$

Then, Bayes' rule is applied for the optimisation of the objective function parameters.

$$P(\alpha,\beta|D,M) = \frac{P(D|\alpha,\beta,M)P(\alpha,\beta|M)}{P(D|M)} \tag{2.52}$$

When a uniform prior density, $P(\alpha,\beta|M)$, is assumed for the regularisation parameters, the maximisation of the posterior is achieved by maximising the likelihood function, $P(D|\alpha,\beta,M)$. On the other hand, it is worth to noting that this likelihood function is the normalisation factor for equation 2.44. As all probabilities have a Gaussian form and equation 2.49 shows the form for the posterior density, equation 2.44 can now be solved for the normalisation factor:

$$P(D|\alpha,\beta,M) = \frac{P(D|\mathbf{w},\beta,M)P(D|\mathbf{w},\alpha,M)}{P(\mathbf{w}|D,\alpha,\beta,M)} \tag{2.53}$$

$$P(D|\alpha,\beta,M) = \frac{\left[\frac{1}{Z_D(\beta)}\exp(-\beta E_D)\right]\left[\frac{1}{Z_{\mathbf{w}}(\alpha)}\exp(-\alpha E_{\mathbf{w}})\right]}{\frac{1}{Z_F(\alpha,\beta)}\exp(-F_{\mathbf{w}})} \tag{2.54}$$

$$P(D|\alpha,\beta,M) = \frac{Z_F(\alpha,\beta)}{Z_D(\beta)Z_{\mathbf{w}}(\alpha)} \cdot \frac{\exp(-\beta E_D - \alpha E_{\mathbf{w}})}{\exp(-F_{\mathbf{w}})} \tag{2.55}$$

$$P(D|\alpha,\beta,M) = \frac{Z_F(\alpha,\beta)}{Z_D(\beta)Z_{\mathbf{w}}(\alpha)} \tag{2.56}$$

where $Z_D(\beta)$ and $Z_{\mathbf{w}}(\alpha)$ are known constants from equation 2.61. $Z_F(\alpha,\beta)$ is the only unknown part and a Taylor series expansion can be used to estimate it. As the objective function is formed in the shape of a quadratic in a small area surrounding a minimum point, $F_{\mathbf{w}}$ can be expanded around the minimum point of $\mathbf{w}^{\mathrm{MP}}$,the posterior density, where the gradient corresponds to zero (MacKay, 1992b,a; Foresee and Hagan, 1997).

$$Z_F \approx (2\pi)^{N/2}(\det((\mathbf{H}^{\mathrm{MP}})^{-1}))^{1/2}\exp(-F(\mathbf{w}^{\mathrm{MP}})) \tag{2.57}$$

Here, $\mathbf{H}$ (second derivatives of the performance index at the current values of the weights and biases) is the Hessian matrix of the objective function and is represented by:

$$\mathbf{H} = \beta\Delta^2 E_D + \alpha\Delta^2 E_{\mathbf{w}} \tag{2.58}$$

If this is placed into equation 2.56, the optimal values of α and β at the minimum point can be solved by taking the derivative with respect to each log and setting them to zero (Burden and Winkler, 2009). This results in the following equations:

$$\alpha^{\mathrm{MP}} = \frac{\gamma}{2E_W(\mathbf{w}^{\mathrm{MP}})} \tag{2.59}$$

$$\beta^{\mathrm{MP}} = \frac{n-\gamma}{2E_D(\mathbf{w}^{\mathrm{MP}})} \tag{2.60}$$

The quantity γ is the effective number of parameters. It measures how many network parameters are effectively used in the reduction of the error function and can range from zero to the total number of parameters in the network, N.

$$\gamma = N - 2\alpha^{\mathrm{MP}}\mathrm{tr}(\mathbf{H}^{\mathrm{MP}})^{-1} \tag{2.61}$$

To optimise the Bayesian regularisation parameters, the calculation of the Hessian matrix of $F(\mathbf{w})$ at the minimum point $\mathbf{w}^{\mathrm{MP}}$ is an essential step in the framework. Foresee and Hagan (1997) proposed the Gauss-Newton approximation to the Hessian matrix. The iterative procedure steps in their method are as follows:

1. Initialisation of β and α and the weight values.
2. Extract one step of the Levenberg-Marquardt algorithm to find the weights minimising the objective function (Hagan and Menhaj, 1994; Foresee, 1996; Hagan et al., 1996).
3. Computation of the effective number of the quantity γ.
4. Gain new estimates for the objective function parameters, $\alpha = \frac{\gamma}{2E_W(\mathbf{w})}$ and $\beta = \frac{n-\gamma}{2E_D(\mathbf{w})}$.
5. Iterate steps 2 through 4 until convergence is achieved.

2.3.4 Comparison of Prognostic Models and Hybrid Applications

As hypothesised by the impossibility theorem of optimisation (no free lunch theorem - NFLT), a general-purpose universal strategy of optimisation is impossible and a strategy can outperform another when it concentrates on the structure of a specific case under consideration (Ho and Pepyne, 2001; Koppen, 2004). Particularly in prognostics, no prognostic algorithm is ideal for every problem (Coble, 2010). A variety of models in the literature have been introduced for application to certain situations or specific domains.

It is a challenging task to estimate the trends of all characteristic parameters by using a single prognostic model since the associated parameters are diverse in real-world applications (Peng et al., 2010). Therefore, it is very common that a prognostic model is integrated with alternative methods based on the system characteristics and the measured condition data at run time; this essentially becomes a hybrid application.

The hybrid category consists of the combination of different prognostic applications (Lee et al., 2014). Table 2.6 describes the extent to which these approaches are compared to each other, and applied in different domains. A hybrid prognostic approach may have more general significance when the desired merits are included, and the drawbacks are removed. Also, it can leverage the strengths of the methods by leading to improved prognostic results.

In addition to the advantages of the approaches that they are based on, the hybrid prognostic approaches include the following several advantages (Eker, 2015).

- The inadequacies caused by individual approaches could be compensated
- A higher prediction accuracy and prognostics performance can be achieved
- The complexity of computations can be reduced

Table 2.6: Comparison of Prognostic Approaches (Eker, 2015; Liao and Köttig, 2014; Sikorska et al., 2011)

	Merits	**Limitations**
Physics Based Models	Less (or no) data requirement	Component (or system) specific
	High precision and accuracy (if a representative mathematical model is available to model the degradation)	Difficulty in system modelling (particularly for complex systems working under multiple operational conditions)
	Suitable to use in design phase of simple systems	Confined to the material (and design) properties due to the requirements for characteristic component information and failure mechanism
Knowledge Based Models	Simplicity in design and modelling No model is required	Limitations in obtaining domain knowledge of new situations (The models mimicking the expert behaviour are more likely to reflect this)
	Computationally less expensive	Requires a data library and multiple training cases to feed the model
	Ability of dealing with multi step ahead remaining useful life estimations	
Data Driven Models	Easy for implementation (less system information is required)	Need to be combined with alternative approaches
	Flexibility and adaptability for different cases	High computational complexity
	Suitable for complex systems	Risks on inaccurate estimations due to change in time

In the field of complex system prognostics, it is anticipated that a specific method is not always accurate or effective for systems formed of interrelated components, and therefore a hybrid method or a fusion approach to prognostics will greatly improve prediction accuracy and capability (Sun et al., 2010). Complex system work under superimposed operational margins and the damage progress of such cases is not deterministic, and usually multidimensional (Saxena et al., 2008b). Therefore, a simple model is unable to present the damage progress and one should consider more advanced techniques (Cempel, 2009).

One of the most prominent of such advanced methods for complex domains is neural networks (Parker Jr et al., 1993; Brotherton et al., 2000; Bonissone and Goebel, 2002; Heimes, 2008; Goebel et al., 2008b; Peel, 2008; Baraldi et al., 2013). These applications are practical for complex engineering systems formed of non-linear dynamical systems and they can model prognostics without knowing the exact relationship between input and target data series. However, a network model requires a target output since the ANNs train itself by the association between input and target data (Cross et al., 1995). The network function can only estimate the desired response and the actual outcome if the network training is supervised by an output representing the degradation model.

To form a supervised network model, there is a need for the feature extraction transformation of data from high-dimensional space to a space of single health level dimension (Bektas et al., 2017). Although the single health level (or alternatively the damage progress) can be produced by pre-defined models such as the generalised time-varying health index equation introduced by Saxena et al. (2008b), there is always a risk to fail in identifying characteristic of individual cases with these models. Statistical feature extraction and dimension reduction models such as linear regression (Chatterjee and Hadi,

1986; Freedman, 2009; Ramasso, 2014a), principal component analysis (Ramasso, 2014a,b; Juesas et al., 2016; Bektas et al., 2017) and multi-regime normalisation (Peel, 2008; Wang et al., 2008; Wang, 2010; Lam et al., 2014; Malinowski et al., 2015; Rigamonti et al., 2016) can standardise multidimensional data to a space of single health level dimension for the use of network target assignment.

There is one further aspect that should be mentioned, and this is directly attributable to the RUL estimation. After the multidimensional data is filtered into amore appropriate form for prognosis, the multi-step ahead prediction needs to be obtained for RUL estimation. In such cases, the predictions are typically faced with growing uncertainties arising from various sources and time series forecasting methods struggle to provide accurate estimations on such problems. The multi-step ahead estimations by ANN models can be quite challenging when little historical data is available and the failure is expected to happen in the longer term (Menezes and Barreto, 2008). The stochastic approaches such as Bayesian networks (Jensen, 1996; Zhang et al., 2007), Particle Filtering (An et al., 2013; Miao et al., 2013; Wang and Gao, 2014), Kalman Filtering (Hu et al., 2012a; Julier and Uhlmann, 1997; Swanson, 2001) and Markov models (Camci, 2005; Baruah and Chinnam, 2005; Dong and He, 2007; Ramasso, 2009) or the statistical time series forecasting models such as trend extrapolation (Batko, 1984; Kazmierczak, 1983; Cempel, 1987) and auto regressive models (Yan et al., 2004; Galati et al., 2008) can contribute to this issue and provide multi-step ahead predictions. However, the issue of historical data shortage is also a problem for these cases and when the prognostic performance along with the risk of catastrophic failures is considered, there is always a requirement to achieve higher prediction performance.

One well-known approach capable of achieving high prediction accuracy is the trajectory similarity-based prognostic approach. The method made the

multi-step predictions over the pairwise distance similarity between filtered outputs (health indicators - a space of single health level dimension). Since the estimation is exclusively based on best-matching past operations and known failure times, the similarity-based model demonstrated significant improvement in performance over other approaches (Wang, 2010). Furthermore, if the method is merged with alternative methods (such as ANN filtering and dimensionality reduction models) to assign health indicators for similarity-based predictions, there would be a noteworthy potential to improve prognostic performance.

To decide the best practice for a hybrid approach, the prognostic performance plays a vital role. Performance evaluation methods help understand the differences and similarities between various methods and what may or may not be borrowed. Further, they may be useful in deciding upon a suitable hybrid approach for a specific application (Saxena et al., 2008a).

2.4 Performance Evaluation

Today, the maintenance management is under pressure to advance their capacities to introduce value-based improvement for the operations (Tsang, 2002). As more research communities start adopting prognostics for maintenance applications, it becomes more necessary to use standardised prognostic techniques (Uckun et al., 2008) as well as the performance measurement metrics (Goebel et al., 2011a). On the other hand, prognostics is a developing field and there are major issues in modelling a prognostics method that can be deployed in field applications (Schwabacher, 2005). This is mainly due to the basic deficiencies on the metrics to properly evaluate the prediction performance (Leao et al., 2008). Therefore, there is always a requirement for prognosis certification in successful deployment of a prognostic system (Saxena et al., 2008a).

This allows to analyse the literature and identify the most suitable algorithm among several. Also, users might have different requirements in their applications; hence the metric can be adopted for each end user for a customer based verification (Jolliffe and Stephenson, 2003). Although this might pose a conflicting need to the idea of generalisation, the metrics can be tailored to model for how to choose metrics for specific cases and also compare a model with other competing ones in the literature (Saxena et al., 2008a).

Prognostic metrics are regarded as a standardised method of communication by which the developers and users demonstrate their results and compare findings (Goebel et al., 2011a; Bektas, 2014). This communication supports the suitable expression of requirements and scientific information. Over the years, as a result of various prognostic implementations in a wide variety of disciplines, the prognostic field has established various sets of metrics to assess forecasting performance such as the works of Saxena et al. (2008a, 2009b), Leao et al. (2008) and Goebel et al. (2011a). These metric sets principally aim to validate the performance of prognostics applications. Since they are mostly focused on applications where run-to-failure data is available and true RUL is initially known, their usage is of particular importance at the model development stage where the metric feedbacks can be used to integrate the prognostic procedures (Goebel et al., 2011a).

Before presenting the mathematical definitions of these metrics and their relationship to the stages of prognostics design, the following terms and notations relating to following equations are defined here to reduce ambiguities that may arise in subsequent sections.

- e is the error term between predicted RUL and actual-time-to failure.

- t is the time index.

- n is the number of RUL estimations.

- EOP (End-of-Prediction) is the earliest time after the prediction value crosses the failure threshold.

- P is the time index at which the first multistep ahead prediction is made by the prognostic model.

- **Error** (George et al., 2006) Error is basically stated as the deviation from a desired target. Most accuracy-based metrics used in this work is derived directly or indirectly from the error term.

$$\text{Error} = e_i = y_i - \hat{y}_i \tag{2.62}$$

where $\hat{y}_i$ is the estimated value (estimated time to failure - ETTF) and y_i is the desired output value (actual time to failure - ATTF). According to this definition, the absolute error is given by:

$$\text{AE} = |e_i| = |y_i - \hat{y}_i| \tag{2.63}$$

- **MAE** (Hyndman and Koehler, 2006)
  Considering that there is more than one instance, the mean absolute error (MAE) calculates an average of the absolute error terms. This quantity is used to measure how close estimations are to the eventual outcomes.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|e_i| = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.64}$$

- **Average Bias (AB)** (George et al., 2006; Saxena et al., 2008a)

  AB is a prognostic metric that averages the prediction errors that occur at all subsequent times after the first prognostic calculation is made. To

establish an overall bias measurement, it can be extended to all operational units under test.

$$AB = \frac{\sum_{i=P}^{EOP} y_i - \hat{y}_i}{(EOP - P + 1)} \tag{2.65}$$

- **Mean Square Error (MSE)** or mean squared deviation (MSD), (Hyndman and Koehler, 2006)

  MSE is a risk function that calculates the average of the squares of the errors. When the vector of predictions is obtained and the vector of known true RUL values is available, the MSE can be estimated by:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}{e_i}^2 \tag{2.66}$$

- **False Positive Rate (FP) and False Negative Rate (FN)** (Goebel et al., 2011a)

  FP is the ratio where a fault is predicted in spite of the asset performing within desired conditions. Conversely, a negative rate is the ratio of unanticipated predictions when the engine would fail.

  Both metrics are at the heart of error-based metrics, and they deserve special consideration in terms of performance validation (Goebel et al., 2011a). Users are required to set acceptable ranges for predictions according to their operational standards. While early in time predictions result in excessive lead time, which may lead to unnecessary corrections, a late in time prediction is a more critical threshold time unit, which might cause catastrophic consequences during the occurrence of any failure.

  Mathematically, they are defined as

$$FP(i) = \begin{cases} 1 & error > t_{FP} \\ 0 & \text{otherwise} \end{cases} \tag{2.67}$$

and

$$FN(i) = \begin{cases} 1 & -error > t_{FN} \\ 0 & \text{otherwise} \end{cases} \tag{2.68}$$

where t_{FP}, and t_{FN} are the user-defined acceptable early or late prediction limits. FP and FN both vary between 0 and 1, where 1 represents the direction of rate score. Their percentage can be compiled to measure the consistency and reliability of prognostic models.

- **Mean Absolute Percentage Error (MAPE)**, or mean absolute percentage deviation (MAPD), (Makridakis et al., 1982; Saxena et al., 2008a) MAPE is a measure of prediction accuracy which averages the absolute percentage errors in the predictions of multiple RUL calculations at the same prediction horizon. It addresses accuracy as a percentage.

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{2.69}$$

Median, in a similar way, could be used to compute the median absolute percentage error (MdAPE).

- **Standard deviation** (Hyndman and Koehler, 2006; Saxena et al., 2008a)

Standard deviation expresses by how much the members of a group deviate from the mean value. It is used to calculate the amount of variation from a set of data. With this definition, a low standard deviation corresponds to those data points being closer to the mean of the set, while a high standard deviation corresponds to the data points expanding over

a broader range of values.

Considering that y takes sequential values from a finite calculated data set $y_1, y_2, ..., y_n$, the standard deviation is expressed by:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \mu_p)^2} \tag{2.70}$$

where μ_p is the mean of population.

$$\mu_p = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{2.71}$$

This type is called the population standard deviation and divided by n to calculate the variance

Sample standard deviation, on the other hand, measures the deviation of the error with respect to the sample mean of the error value, and it is restricted to the assumption that the error has a normal distribution.

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \mu_e)^2} \tag{2.72}$$

The deviation is divided by $n - 1$ when calculating the variance. The sample mean of the error in this equation corresponds to;

$$\mu_e = \frac{1}{n}\sum_{i=1}^{n} e_i \tag{2.73}$$

- **Mean absolute deviation from the sample median (MAD)** (Saxena et al., 2008a)

  This measurement is a resistant estimator of the variability of the prediction error. It refers to the population parameter, which is estimated

by the mean of the median absolute sample deviations calculated from a data sample.

$$MAD_i = \frac{1}{n}\sum_{i=1}^{n}|e_i - M| \tag{2.74}$$

$$M = median(e) \tag{2.75}$$

The corresponding formula for median values separating the upper half of the input variables in the same time step from the lower half can be expressed by the following equations (Hogg and Craig, 1995).

According to the order statistics:

$$x_{f_1} = \min_j \tilde{y}_{tj}, x_{f_2}, , x_{f_{N-1}}, x_{f_N} = \max_j \tilde{y}_{tj} \tag{2.76}$$

The statistical median of the input variables at time step j is defined by:

$$y_{tj} = \begin{cases} x_{f_{(n+1/2)}} & \text{if } n \text{ is odd} \\ \frac{1}{2}\left(x_{f_{n/2}} + x_{f_{1+n/2}}\right) & \text{if } n \text{ is even} \end{cases} \tag{2.77}$$

- **Median absolute deviation from the sample median (MdAD)** (Saxena et al., 2008a) In a similar way to the MAD equation, MdAD calculates the median deviation from the sample standard deviations calculated from a data sample.

$$\text{MdAD} = median\left(|e_i - M|\right) \tag{2.78}$$

$$M = median(e) \tag{2.79}$$

- **Score Function** (Saxena et al., 2008b)

  Performance evaluation is mainly concerned with using accurate metrics that help assessing whether the prognostic algorithms meet requirements for the task at hand. For a critical degradation scenario, an early or late prediction may be preferred over another. The "Score Function" is designed to satisfy this requirements. It is a metric score of estimated calculations and a weighted sum of RUL errors (Saxena et al., 2008a), Goebel et al. (2011a). In either an early or late case, the penalty grows exponentially with increasing error. The asymmetric preference is controlled by user-defined acceptable early and late parameters, a_1 and a_2, in the scoring function described below:

$$s = \begin{cases} \sum_{i=1}^{n} e^{-\left(\frac{d}{a_1}\right)} & \text{for} \quad d < 0 \\ \\ \sum_{i=1}^{n} e^{\left(\frac{d}{a_2}\right)} & \text{for} \quad d \geq 0 \end{cases} \tag{2.80}$$

  where s is the computed score, n is the number of predicted units, and d is the error term.

  In prognostics context, the key aspect is to avoid catastrophic failures and it is intended to predict the RUL early as compared to predicting late. However, in certain cases where the failures may not pose catastrophic or life-threatening situations and early predictions may instead result in undesired economic costs. Accordingly, the asymmetric preference of the "Score Function" may change and one may not prefer conservative predictions. Therefore, a performance evaluation model based on this equation should reflect such preferences to meet specific

requirements (Saxena et al., 2008b).

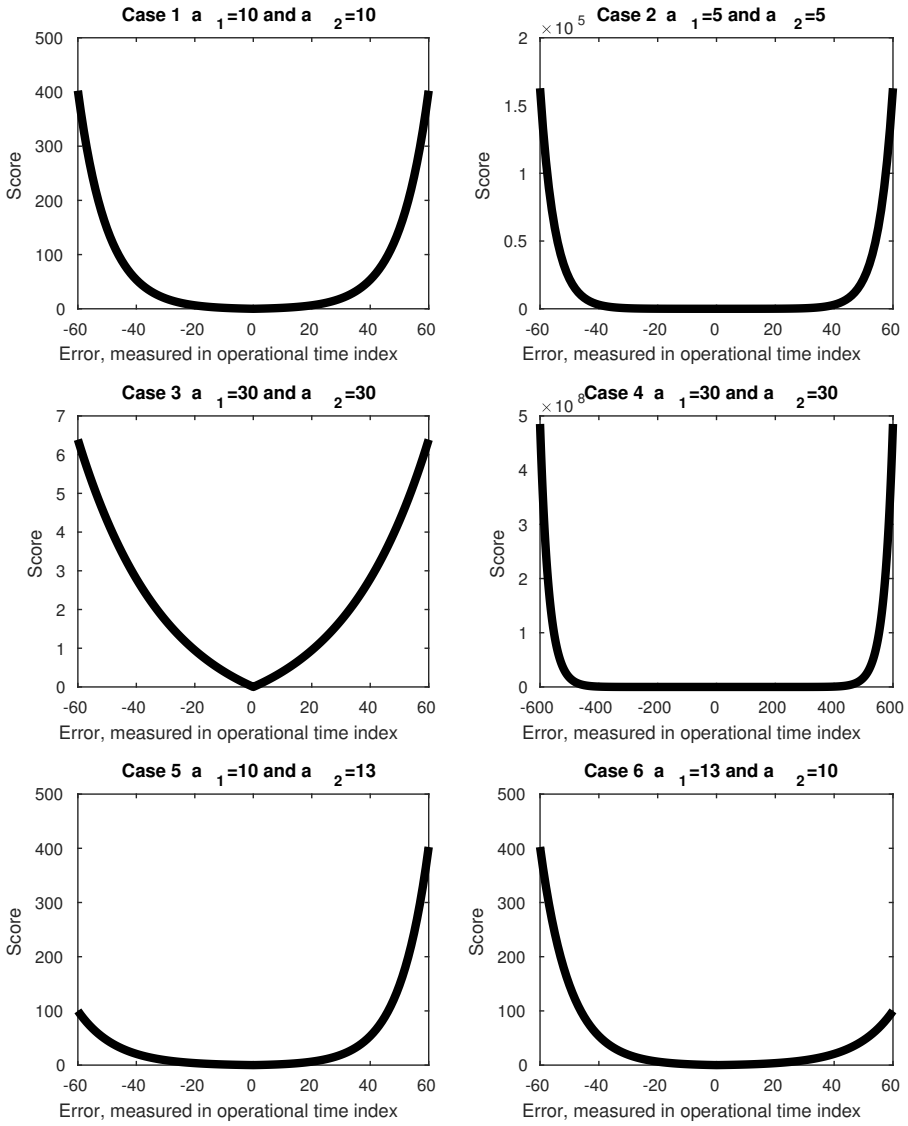

Figure 2.9: Characteristic of Scoring Function

Figure 2.9 shows different score scenarios as a function of the error. In Case 1 and 2, the parameters a_1 and a_2 are equal and the preference is symmetric. However, both cases increases exponentially with error rate. Exponential growth of the score is exhibited when the rate of error change measured in operational time index of the equation 2.80 is

proportional to the function's current value, resulting in the scoring value at any time being an exponential function of time index, ie, a scoring function in which the error value is a part of the exponent. As this error value is divided by user-defined acceptable early and late parameters, the magnitude of score is dependent on the relation between the length of operational time index and the exponent term.

The important aspect to include here is the magnitude of user-defined parameters. If the length of condition monitoring time series is short, the exponent term should be assigned with higher values and accordingly, the user-defined parameters should be assigned with lower values. However, when the time series is relatively large, the parameters should be assigned higher to increase the performance evaluation. Otherwise, the score might be biased to error growth as in Case 2. A further demonstration of the relationship between user-defined parameters and the error size is shown in Case 3 and 4. Considering that the time index is measured by operational cycles and time series are short (say less than 300 units), the user-defined parameters assigned to "$a_1 = a_2 = 30$" as in Case 3 cannot exactly exhibit the non-linear results desired from the scoring function in which lower error rates are preferred to higher ones. Conversely, if the time index is measured by more frequent cycles such as days or weeks, and time series are longer, the function over scores the results as in Case 4.

Saxena et al. (2008b) first introduced this function for cases where each data point is a snapshot of an entire operational cycle so that the run-to-failure time series are mostly less than 300 time steps and the late predictions are considered more punishable than the early predictions. Case 5 shows their scenario with an asymmetric preference; however one may instead require to consider the undesired economic costs (early

predictions) as in Case 6.

The prognostic performance metrics explained in this section are commonly applied to provide an accurate validation of RUL estimation models. Since a correct evaluation is required to feed the prediction design, their use could significantly contribute to the structure of the prognostics. In complex systems particularly, the accuracy measured by metrics represents the closeness of prediction measurements to the true value of system health conditions.

2.5 Challenges of Prognostics

It was reported by Wang (2010) that successful prognostic applications are still rare in complex engineering systems, even though many algorithms have been proposed and experimented for RUL calculations. Multiple misconceptions and issues encountered during the development of such algorithms are challenging to prognostic applications in complex systems. Moreover, the complex data characteristics exhibit the immensely stochastic and nonlinear degradation patterns that make the system difficult to model accurately (Eker, 2015). Prognostics, therefore, are considered the Achilles' heel of PHM systems and, since they entail large-grain uncertainty, the prognostic discipline presents major challenges to the condition-based maintenance applications (Roemer et al., 2006).

The prognostic challenges, and their potential requirements, observed in the current literature are given in the following subsections.

2.5.1 Lack of Common Data Sources

The development of advanced prognostic techniques is an area of active research, and a successful model requires continuous collection of data during

the lifetime of an asset. As operations progress, there will be evolving, changing classes of asset faults and hence accompanying estimations in prognostic systems are needed to detect them (Brotherton et al., 2002). A barrier to prognostic methodology studies is the availability of common case studies and data sets for the development and validation of algorithms (Pecht, 2008).

As mentioned in section 2.1.1, the common data sets, by which researchers can develop and compare their methods, are necessary for a successful data acquisition stage in CBM and are an important part of prognostics in general.

2.5.2 Data Characteristics

For prognostic applications, it is almost impossible to comprehend and model every detail of the behaviours in a complex system, and what is presented is generally limited to the overall operating system mechanism and some micro-level physics models of critical system components (Wang, 2010). Therefore, the calculations and projections are prone to substantial uncertainty across a large number of operational inputs. Since each instance in a dataset has characteristic features such as initial wear level, failure mode and operating conditions (Saxena et al., 2008b), the population of entire cases should be taken into account when the signal processing is applied in order to receive useful health information for projections. A dataset standardised into a common scale can provide higher accuracy but, when novel data is introduced, the population parameters of the dataset change and the signal processing needs to be repeated to consider any new instances.

2.5.3 Uncertainty in Predictions

Operational conditions and different regimes have a considerable impact on system degradation behaviour, noise, uncertainty and errors in multidimen-

sional data. The common data-driven prognostics are based on the assumption that historical operational conditions can facilitate a reasonable model for RUL estimation, whereas future operational conditions are unknown and need be projected. It is, therefore, accepted that a system's future operational behaviour or usage will follow an estimated pattern found from historical training data. However, when the test data length is short and long-term projections are necessary, the prognostic accuracy may fail to provide desirable results. Even if the future of short test data actions can be projected, the error between actual time-to-failure and estimated time-to-failure could be catastrophic in terms of CBM requirements. Therefore, prognostic prediction requires a solution that has a high probability of identifying training failure degradation and representation of future steps.

2.5.4 Validation Issues

RUL prediction is different from future behaviour predictions that are validated after the system has experienced a whole life cycle and come to a real failure point; this process can take a very long time for many engineering systems. If the actual time-to-failure is known initially, the prediction performance can be validated by the different metrics introduced in section 2.4. Metrics developed in the context of prognostics are a common validation method and users can compare their findings and results. Moreover, these metrics can address any primary assessment of algorithmic performance in prognostic applications, and they are mainly focused on tackling offline performance evaluation methods for applications where true RUL is known a priori. Therefore, they are especially useful at the algorithm development stage in which metric feedbacks can be used to fine-tune prognostic algorithms (Goebel et al., 2011a). However, short length test data instances are particularly risky in terms of error-based metrics. If an operation is interrupted by an unexpected

fluctuation caused by operational conditions, the projection is prone to deviations which affect overall results.

2.6 Expected Contributions to the Literature

As discussed in section 2.3, different prognostics approaches have various requirements to capture the degradation process and estimate the RUL of a system. Each prognostic method has its own merits and limitations, and a hybrid approach can integrate any desired advantages whilst avoiding characteristic disadvantages.

This research intends to identify a multifaceted prognostic model consisting of data processing and multi-step ahead predictions. The model is particularly based on prognostics of complex systems, and seeks an adaptive solution for multidimensional data under various regimes. The model will firstly adapt the common dataset resources and prognostic methods described in the literature, and then it will develop an adaptive filtering and prediction model with due consideration to prognostics metrics, which make the prognostic performance and accuracy more effective.

A framework for neural network based filtering and trajectory similarity-based estimation will be introduced to calculate the RUL estimations. Then, the model can be applied to well-known prognostic datasets, and the performance with these datasets could then be compared to existing models in the literature.

The review of the literature in this chapter shows that the current methods lack the ability to deal effectively with the multi-dimensional data and lack of collaboration between different but similar data sources. Further traditional pre-processing is unable to deal with this efficiently. This research addresses these shortcomings and contributes to the literature by introduc-

ing an adaptive filtering model that can standardise different trajectories in a dataset without requiring population parameters. The standardised outputs are on a common scale and can be used for collaborative multi-step projection. Since the data characteristics of different trajectories are taken into account, the model proposes a generic prognostic integration scheme where multiple models are integrated into the different phases of prognosis.

Chapter 3

Methodology

This chapter first provides background information on prognostics requirements for complex systems working under different operational conditions. The framework of practised methods is then justified, and the development of a hybrid approach for the multidimensional data and complex systems is laid out. The architecture and working procedures of the proposed techniques and theories are described in detail. The approaches are discussed under three main categories: dimensionality reduction, network filtering modelling and RUL estimation.

3.1 Motivation behind the Methodology

The difficulties in accurately and precisely predicting RUL have led to a wide variety of methodologies and algorithms in the prognostic literature, which were discussed in Chapter 2. However, the review of the literature reveals that the existing prognostics development methods have not always well thought out when dealing with complicated interdependency (Günel et al., 2013), multidimensional failure mechanism (Saxena et al., 2008b) and noisy data (Uckun et al., 2008). Due to the lack of understanding in these issues, the current

literature is not able to fully utilise condition monitoring histories and to integrate population reliability information into RUL prediction, and to derive the relationship between the health index and condition measurements (Heng, 2009).

This chapter presents a novel prognostic framework to address these unresolved shortcomings. The work particularly focuses on the issues of prognostics of specific complex systems which operates under various conditions and provides condition monitoring histories with multidimensional failure mechanism and data. The main motivation is to provide a data processing-oriented RUL prediction method by using various system indicators and sensor measurements. The fundamental research inquiry to be analysed in the methodology is the interpretation of data pre-processing techniques to aid the performance of prognostic applications. The methodology has developed a conceptual prognostic model to overcome the challenges presented by multi-dimensional data.

3.2 Problem Definition

Multidimensional data (and also multidimensional failure mechanism) are represented in terms of data dimensions, which are arranged in dimension hierarchies Vassiliadis (1998). To model the space of such multidimensional data, let $\mathbf{\Omega}$ be the space of all dimensions. An operational case ($\mathbf{T}$) in this space is formed of a set of time $\mathbf{t}$, operating conditions (or more commonly known as regimes) $\mathbf{r}$ and the multidimensional condition monitoring data $\mathbf{x}$.

$$\mathbf{T}_{(i)} = \langle \mathbf{t}_i, \mathbf{r}_i, \mathbf{x}_i \rangle \,, \tag{3.1}$$

$$i = 1, 2, \cdots, n, \text{and}, \mathbf{T}, \mathbf{t}, \mathbf{r}, \mathbf{x}, \in \mathbf{\Omega} \tag{3.2}$$

This is written in matrix form as follows:

$$\mathbf{T} = \left\langle \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ \cdots \\ t_n \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \cdots \\ r_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ x_n \end{pmatrix} \right\rangle \tag{3.3}$$

where $\mathbf{t}$ follows a sequential order and each time index variable is unique.

$$\mathbf{t} = (t_1, t_2, t_3 \cdots t_n) \tag{3.4}$$

$$\theta(t) = \{t_i\}_{i\epsilon\{1,2,3,\cdots,n\}} \tag{3.5}$$

$$|\theta(t)| = n \tag{3.6}$$

where $\theta(t)$ represents the number of unique time index variables and n is the number of steps in the time series of the operational trajectory ($\mathbf{T}$).

On the other hand, the complex system under dynamic working regimes operates at superimposed operational margins at any given time instant and the variable of operating conditions, $\mathbf{r}$, can only have certain values (Goebel et al., 2007; Rareshide et al., 2009; Uluyol et al., 2011; Goebel et al., 2011b). Therefore, the unique values of the operating conditions vector, $\theta(r)$, can only have certain numbers and are limited to the regimes, np.

$$\mathbf{r} = (r_1, r_2, r_3, \cdots r_n) \tag{3.7}$$

$$\theta(r) = \{r_i\}_{i\epsilon\{1,2,3,\cdots,n\}} \tag{3.8}$$

$$\theta(r) = (1, 2, \cdots, np) \tag{3.9}$$

and

$$\forall r_i, r_i \in \theta(r) = (1, 2, \cdots, np) \tag{3.10}$$

The sensors under different operating conditions provides similar in degree readings with those in the same operating condition (Diallo, 2010). Considering these similarities, the vector, $\mathbf{x} = (x_1, x_2, x_3, \cdots x_n)$, would cluster in certain operating condition domains, ϑ_{r_i}, which are bounded by upper limits u_{r_i} and lower l_{r_i}.

$$\vartheta_{r_i} = (l_{r_i}, u_{r_i}) \Rightarrow \{x_i \in \mathbf{\Omega} : l_{r_i} \leq x_i \leq u_{r_i})\}, \tag{3.11}$$

$$\text{where,} \quad i = (1, 2, \cdots n) \quad \text{and} \quad \forall r_i, r_i \in \theta(r) = (1, 2, \cdots, np) \tag{3.12}$$

Due to the environmental conditions and the characteristics of condition monitoring, such raw data generally contains noise and measurement uncertainties, w, along with the health information, H. Considering the regimes and domains in which the systems operates, the condition monitoring vector is redefined as:

$$x_n^r = \mathrm{H}_n^r + w_n^r \tag{3.13}$$

$$r = (1, 2, \cdots, np) \tag{3.14}$$

$$\forall \mathrm{H}_n^r, \quad \mathrm{H}_n^r \in \vartheta_r \quad \& \quad l_r \leq \mathrm{H}_n^r \leq u_r \tag{3.15}$$

$$\forall w_n^p, \quad w_n^p \in \vartheta_p \quad \& \quad l_p \leq w_n^p \leq u_p \tag{3.16}$$

where it is commonly assumed that w_n^p is drawn from a Gaussian white noise that gives a constant power spectral density (Li et al., 2000; Menezes and Barreto, 2008; Lam et al., 2014).

Figure 3.1 shows a sample of such multidimensional trajectory. The

data represented here is only for illustration and it is formed of a randomly generated set of cell data, x, and randomly sorted np regime levels. The sensor measurements aligns in various operating conditions (regimes) and condition monitoring measurements.

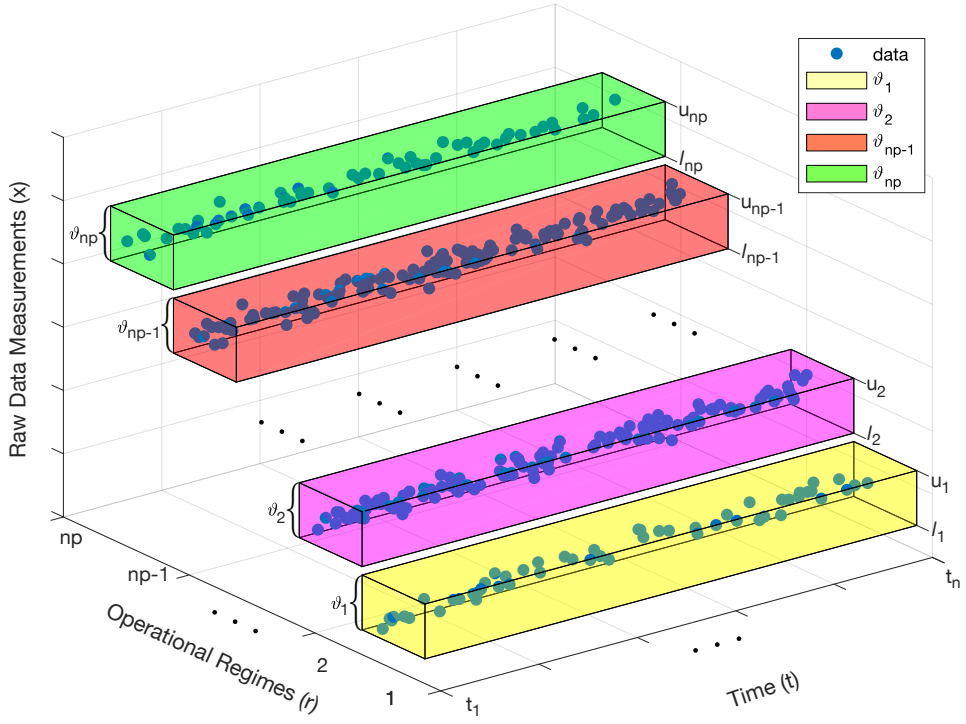

Figure 3.1: Multidimensional Data

Since the science of prognosis is predicated on system ageing and monotonic damage accumulation, it should be possible to correlate sensor behaviour with signs of ageing to estimate the RUL of systems (Uckun et al., 2008). However, the multi-dimensional data such as the one shown in Figure 3.1 could not provide useful information to measure the monotonic damage accumulation. Further applications are needed to provide useful information for RUL predictions.

When it is supposed that the system is instrumented at a certain time index through its life cycle, the reading of the feature vector x_i is timestamped by the time index t_i (Wang, 2010). Let $t_{current}$ is the time stamp of the

current measurement index and t_{end} is the ending time stamp. Then, the RUL estimation problem for a system can be defined as:

$$RUL = t_{end} - t_{current} \tag{3.17}$$

In the case that the run-to-failure data is available from training instances, it is desirable to include their historical condition monitoring data for RUL estimation of test instances which have only instrumented up to the current measurement index $t_{current}$.

- The run-to-failure time index of condition monitoring feature vectors from the test instance includes:

$$t_{test} = [t_1, t_2, \cdots, t_{current}] \tag{3.18}$$

- The run-to-failure time index of condition monitoring feature vectors from the test instance includes:

$$t_{train} = [t_1, t_2, \cdots, t_{end}] \tag{3.19}$$

The prognostic problem in such cases is to effectively use the training instances for RUL estimation of test instances. As for the specific domains for this research (complex systems operating under variable operating conditions), it is indispensable to employ data filtering on the operating conditions of both the training instances and test instances.

3.3 Justification for the Methodology

As prognostic technology advances to anticipate and respond to real-life applications, the desire for using advanced data-driven processing methods has

considerably risen. Data-driven prognostics are mainly practised for modelling of the desired system target with the available historical condition data Kan et al. (2015). Nevertheless, there is an issue of data filtering in data-driven processing methods and it arises when a set of operational cases with multi-regime operating conditions and different initial health levels is inserted into the same data-driven filtering model. This is more evident in a common data in which there are multiple operational cases involved.

Ensuring performance in such cases is a major prominent problem and it requires the theoretical frameworks more applicable to real industrial domains Boussif (2016). As mentioned in section 3.1, in addition to RUL prediction for such systems, a multi-class data processing framework needs to be defined for a multidimensional feature space.

To define such a framework and produce meaningful information from multidimensional space, Peel (2008); Wang et al. (2008) introduces a component-wise "multi-regime normalisation" method to normalise the multidimensional according to each other within the same domain (regime). In their works, the normalisation is applied into a common data which is formed of multiple operational cases with distinct health levels (and also initial performance levels) that can be found in the condition monitoring data. The health levels in these operational cases evolve with exponential characteristics (Saxena et al., 2008b).

$$h(t) = 1 - d - \exp\left\{at^b\right\} \tag{3.20}$$

where d is a case specific initial wear level point in the wear-space (each case is observed with some non-zero initial wear degradation that is unique for each observation), a and b are model parameters which are also case specific. In such a situation, each operational case starts at a distinct operational performance level, "d", and maintains a distinctive "h" pattern. As one should

consider multiple cases and trajectories in a real-life scenario, the entire data with its all components needs to be normalised together to preserve the characteristics of the operational cases.

For the conditions considered in the works of Peel (2008); Wang et al. (2008), all operational cases were available at the same time and provided before the design of the methodologies. The "normalisation of all operational cases at once" has therefore not been a problem to be tackled. However, it would be improbable to follow such a scenario in a real-life application due to the confidentiality and restrictions on data (Ramasso and Saxena, 2014). When the "multi-regime normalisation" is considered in a real-world application, it should be reiterated for each incoming novel operational case to calculate the changing population parameters and the operational case characteristics of h and d.

In other statistical feature extraction and dimension reduction models such as linear regression (Chatterjee and Hadi, 1986; Freedman, 2009; Ramasso, 2014a), principal component analysis (Ramasso, 2014a,b; Juesas et al., 2016; Bektas et al., 2017) and z-test (Daigle et al., 2010; Rigamonti et al., 2016, 2017), the multidimensional data could be normalised to a space of single dimension. However, similar to the "multi-regime normalisation", these also lack the ability to deal with standardisation with regard to all existing operational cases and potential novel ones.

An approach that can be a challenging alternative to the "multi-regime normalisation" and other feature extraction and dimension reduction models is the "neural network filtering" model (Heimes, 2008; Greitzer et al., 1999; Fink et al., 2014; Wu et al., 2016; Loutas et al., 2017; Elforjani, 2016; Yang et al., 2016; Zheng et al., 2017). Unlike the above-mentioned regime normalisation and dimensionality reduction methods, a neural network function which is trained for a single operational case can filter the necessary degra-

dation information for other existing and novel cases. This network function forms a set of interconnected functional relationships between many input series and a desired unique target where the relationship can be trained for optimal performance (Byington et al., 2004a). Since a complex system is formed of interacting components (Jamshidi, 2008), there are multiple sensors for constituents to receive condition monitoring information on the system health or any potential problems (Günel et al., 2013) and these multiple corresponding measurements can be used to train the neural network function with multi-input series. Therefore, the multidimensional condition monitoring data defined in equation 3.2 and 3.3 becomes to multivariate data and there are multiple sensor readings at each time step (say ns sensor readings at t for the following example) .

$$\mathbf{x} = [x_1, x_2, x_3, \cdots, x_t, \cdots, x_n] \tag{3.21}$$

$$\mathbf{x_t} = [x_{t,1}, x_{t,2}, x_{t,3}, \cdots x_{t,ns}] \tag{3.22}$$

Since the monitoring data is not defined as a vector any more, it is written in the matrix form as follows:

$$\mathbf{x} = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,ns} \\ x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,ns} \\ x_{3,1} & x_{3,2} & x_{3,3} & \cdots & x_{3,ns} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & x_{n,3} & \cdots & x_{n,ns} \end{bmatrix} \tag{3.23}$$

A simple model would be unable to present the damage progress in such multidimensional and multivariate input, and one needs to consider more advanced non-linear dynamical systems which can model prognostics without

knowing the exact relationship between input and target data series (Cempel, 2009). Therefore, neural networks have become the most used model in prognostics (Parker Jr et al., 1993; Brotherton et al., 2000; Bonissone and Goebel, 2002; Heimes, 2008; Goebel et al., 2008b; Peel, 2008; Baraldi et al., 2013). An ANN is a convenient algorithm for modelling complex system without understanding the complete relationship between the raw condition monitoring input, x, and the desired target data, y (Murata et al., 1994).

$$y = f_{NN}(x) \tag{3.24}$$

However, this function can only estimate the desired outcome if it is initially supervised by an output representing the target data. This supervised learning task of inferring the function from "labelled historical condition data" also carries the potential risk of failing to identify data population characteristic such as initial wear levels "d" and "h" pattern. Without accurately specifying these with regard to each operational case, the network filtering may be thrown off because the initial bias may disrupt the sensitiveness of RUL estimations.

By considering both the population-wise multi-regime normalisation methods and the neural network models, an initial unsupervised pre-training phase that estimates hidden system degradation behaviour can be employed in a more generic filtering process. In this way, the characteristic degradation parameters in the multidimensional (and also multivariate) data can be identified as desired, and the *"normalisation of all operational cases at once"* issue can be resolved by a supervised neural network data filtering approach. The RUL of an operational test case can be estimated via the known life-time of similar complete examples so that the final life-time estimations can be collaboratively performed by using multiple historical operational cases.

The novelty presented in the methodology is to perform the target pa-

rameters assignment (unsupervised learning) and data filtering (supervised learning) steps sequentially. The proposed hybrid filtering model standardises operational cases to form an output, y, for the network filtering. The main hypothesis for the proposed framework is that there is the possibility that the novel operational cases can be filtered independently even after the training the network function. In accordance with this hypothesis, it is also expected that the proposed hybrid model will be more efficient for the multi step RUL estimations.

Following the feature extraction from multidimensional data for prognosis, the last aspect of the methodology is about multi step ahead prediction. RUL estimations should be able to deal with time series forecasting to provide accurate results. The physics-based models would be inadequate for estimations as they are component specific, computationally expensive to develop and the specific domain experts are needed (Zaidan, 2014). The availability of an expert is also an issue for the most of knowledge-based prognostics such as fuzzy logics (Feng et al., 1998; Satish and Sarma, 2005; Dmitry and Dmitry, 2004) and expert systems (Butler, 1996; Biagetti and Sciubba, 2004). This is more evident especially when there are novel operational cases that are not covered explicitly by an expert.

The data-driven prognostics can provide the multi-step ahead time series forecasting and estimate the RUL such as Bayesian networks (Jensen, 1996; Zhang et al., 2007), Particle Filtering (An et al., 2013; Miao et al., 2013; Wang and Gao, 2014), Kalman Filtering (Hu et al., 2012a; Julier and Uhlmann, 1997; Swanson, 2001), Markov models (Camci, 2005; Baruah and Chinnam, 2005; Dong and He, 2007; Ramasso, 2009), ANN (Bektas and Jones, 2016), trend extrapolation (Batko, 1984; Kazmierczak, 1983; Cempel, 1987) and auto regressive models (Yan et al., 2004; Galati et al., 2008)). However, it can be quite challenging when there is only little previous condition monitor-

ing data available, the degradation is not mature enough to be predicted, and the failure is expected to happen in the longer term. One other major problem in data-driven prognostics is to avoid the imbalance in multi-step ahead predictions for exponential time series in which the models overly imitate the training data (Bektas and Jones, 2016).

In complex engineering conditions where typical prognostic methods are hard to implement and inefficient, a well-known approach for its ability of achieving high prediction accuracy is the similarity-based prognostics (Wang et al., 2008; Wang, 2010; Eker et al., 2014; Lam et al., 2014; Ramasso, 2014a,b; Bektas et al., 2017).

The method makes the multi-step predictions over best-matching past operations and known failure times. Unlike other knowledge-based methods, the similarity-based approach does not need a specific domain expert but it requires sufficient run-to-failure data to form a library to estimate RUL. Since the pre-processing stages of "multi-regime normalisation" and " neural network" filtering can provide estimation from multiple historical operational cases within the same domain, a library of run-to-failure trajectories can be formed for similarity estimation. In the case that these trajectories include sufficient information, the model can be altered to calculate the best-matching stages at any point in their history rather than calculating the initial similarities. Therefore, if the method is merged with alternative methods of regime normalisation and ANN filtering, there would be a noteworthy potential to estimate the RUL of complex systems which works under multiple operational conditions.

3.4 Research Procedures

The research process follows a hybrid framework of prognostic techniques which are particularly developed to overcome the challenges presented by

multi-dimensional condition monitoring data and RUL estimation problem of complex systems. As shown in Figure 3.2, the methodology includes two main sequential stages:

- **Feature Extraction** (Existing Failure Mode): identifies the symptom relations regarding the health index. A step-by-step approach is proposed to identify the existing damage progress in multidimensional data.
- **Multi Step Ahead Prediction** (Future Failure Mode): estimates the RUL, most likely future health index, and its influence factors.

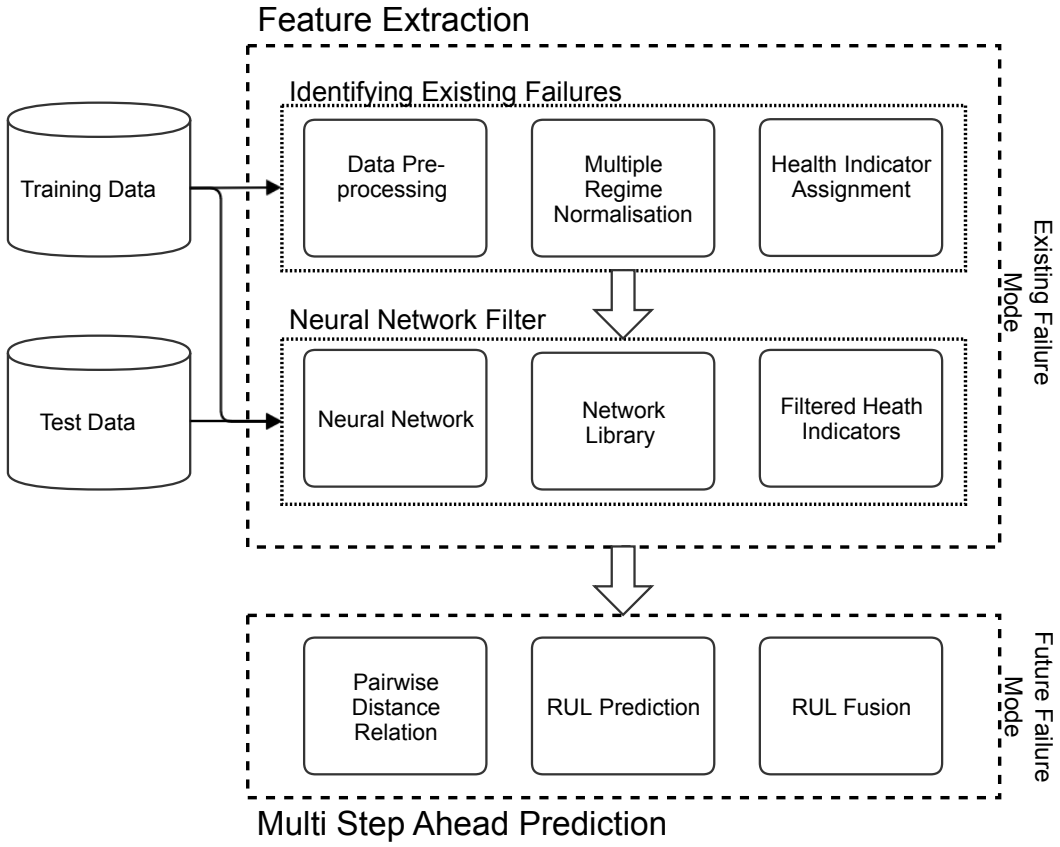

Figure 3.2: Stages of the Methodology

This framework includes a transition from unsupervised regime normalisation to neural network fitting and multi-step-ahead RUL calculations. There

are three major points supporting and assessing the performance of the prognostic structure: multi-regime normalisation, feed-forward neural network and similarity-based estimation. While the multi-regime normalisation and neural network filtering mainly focus on obtaining useful degradation information, the similarity-based estimation deals with the multi-step-ahead prediction of RUL.

One further aspect that can be added to these major points is the post prognostic actions and performance evaluation metrics. While the prognostic algorithm is designed through the application of different procedures for multi-step-ahead RUL predictions, the prognostic metrics in section 2.4 are used dynamically to measure the performance of the model. These metrics and their impact on the technical prognostic requirements have a particular importance in the way that they complement each other. Therefore, this research will be based on both the modelling and performance evaluation of prognostics in terms of technical metrics.

3.4.1 Identifying Existing Failures

Designing a successful prognostic implementation on complex systems requires careful consideration of identification of existing failures as well as the system functionality and operational conditions. Owing to its ability to analyse condition monitoring data, this feature of existing failure mode is a particularly useful analytical tool for development and testing of methodology. It can affect all phases of the prognostic lifecycle prediction and a data analysis that has not been carefully partitioned can produce misleading outcomes. Therefore, the identification of existing failures in multidimensional data is the first and foremost step required before running the RUL calculations. Starting from the initial raw condition monitoring information as described in section 3.2, the following sections describe a mechanism of data processing that, when exe-

cuted properly, produces useful health indicators for neural network filter and similarity-based prediction.

3.4.1.1 Data Pre-processing

Pre-processing of raw data is an essential step to any study relying on any type of data-driven techniques (Diallo, 2010). In the context of this work, in order to develop the multi regime normalisation method and to achieve a useful health index for prognosis, a data processing is applied for an understanding of feature extraction, data cleaning and feature selection. Feature extraction starts from the initial raw data and builds derived values intended to be informative and non-redundant for prognosis, facilitating the subsequent RUL estimations. Feature extraction is related to data cleaning which is the process of identifying and possibly correcting corrupt or inaccurate measurements from the data and requires to gather knowledge on the field to identify inaccurate or irrelevant parts for modification or standardisation (Milano et al., 2005). Then, the process of feature selection is performed to select the relevant features for use in RUL estimation construction.

In real-world data, it is unlikely to find the best attainable high precision model without a validation process (Hernández and Stolfo, 1998). Such data are often inconsistent, incomplete and lacking in certain behaviours, and are likely to contain many errors. Data preprocessing involves transforming these data into an understandable format. Therefore, the methodology first deals with the issues relating to organising the condition monitoring information such as to reduce data redundancy and improve data integrity. The sensors operating under different operational conditions are related to each other, and so the common behaviour of each sensor can be observed and investigated. Then, a health index provides comparable, relevant and actionable information about the sensor population health, as well as tracking progress

and performance over time. At this stage, one is only concerned with training data because they are formed of full operational periods and can demonstrate their initial wear levels together with the failure points. The test data, on the other hand, is limited and will present difficulties in model identification.

3.4.1.2 Multi-regime Normalisation

Considering the condition monitoring data as described in equation 3.11, 3.12 & 3.23, the raw values are multivariate, inconsistent with each other and operating under different regimes. The data need to be adjusted from their original scales to a relatively common scale. A normalisation method (for example the z-score) can carry out these adjustments by returning raw values into a single domain. Z-score standardisation results in a single dimension quantity achieved by subtracting the population mean from each raw value in the same population and then dividing this difference by the population standard deviation (Jain and Dubes, 1988).

The calculation of a z-score for each working regime needs to consider the mean and standard deviation of the regime to which a value point belongs. The equation to calculate the z-score (standard score) of a raw value in a particular regime is given as:

$$N\left(x^r\right) = \frac{x^r - \mu^r}{\sigma^r}, \forall r \tag{3.25}$$

where, x^r is the raw data at regime r, μ is the population mean, σ is the population standard deviation at the same regime feature. Since the data (at the regime r) are formed by n^r scalar observations, the population standard deviation (of r) is:

$$\sigma^r = \sqrt{\frac{\sum_{i=1}^{n^r}(x_i^r - \mu_i^r)^2}{n^r}} \tag{3.26}$$

and the population mean (of r) is:

$$\mu^r = \frac{1}{n^r} \sum_{i=1}^{n^r} x_i^r \tag{3.27}$$

After this equation is applied to all regimes separately, the standardised sensors are used to reassemble the normalised dataset with due consideration to the sensor positions at the initial stage so as to allow them to be assigned on a common scale while maintaining the initial wear characteristics within each trajectory. If the raw time series were reformatted in a given array such as a unity-based feature scaling normalisation method, all values would be transformed into a pre-arranged range (Aksoy and Haralick, 2001). The main risk in such a method that there cannot be a healthy consideration of the initial wear levels and the regime differences. The standard score is, therefore, a robust tool to normalise the data with characteristic features and errors when population parameters are known.

Figure 3.3 shows a sample z-score grading method in a normal distribution and compares it with standard deviations, cumulative percentages and percentile equivalents. A key point in "multi regime normalisation" and "identifying existing failure" modes is that calculating "the standard score" needs the determination of each regime's unique population mean and standard deviation, not merely the entire time series' parameters. Therefore, a function is required to apply the z-score normalisation to all sensors in the same regime instead of the sample sensors drawn from individual trajectories. This will allow for the entire dataset's simultaneous normalisation.

In certain cases where it is possible to measure every sensor in a trajectory, the standard score normalisation is computed by:

$$\hat{x}_{(c)} = \sum_{i=1}^{np} \frac{x_{(c)} - \mu_{(c)}}{\sigma_{(c)}} \tag{3.28}$$

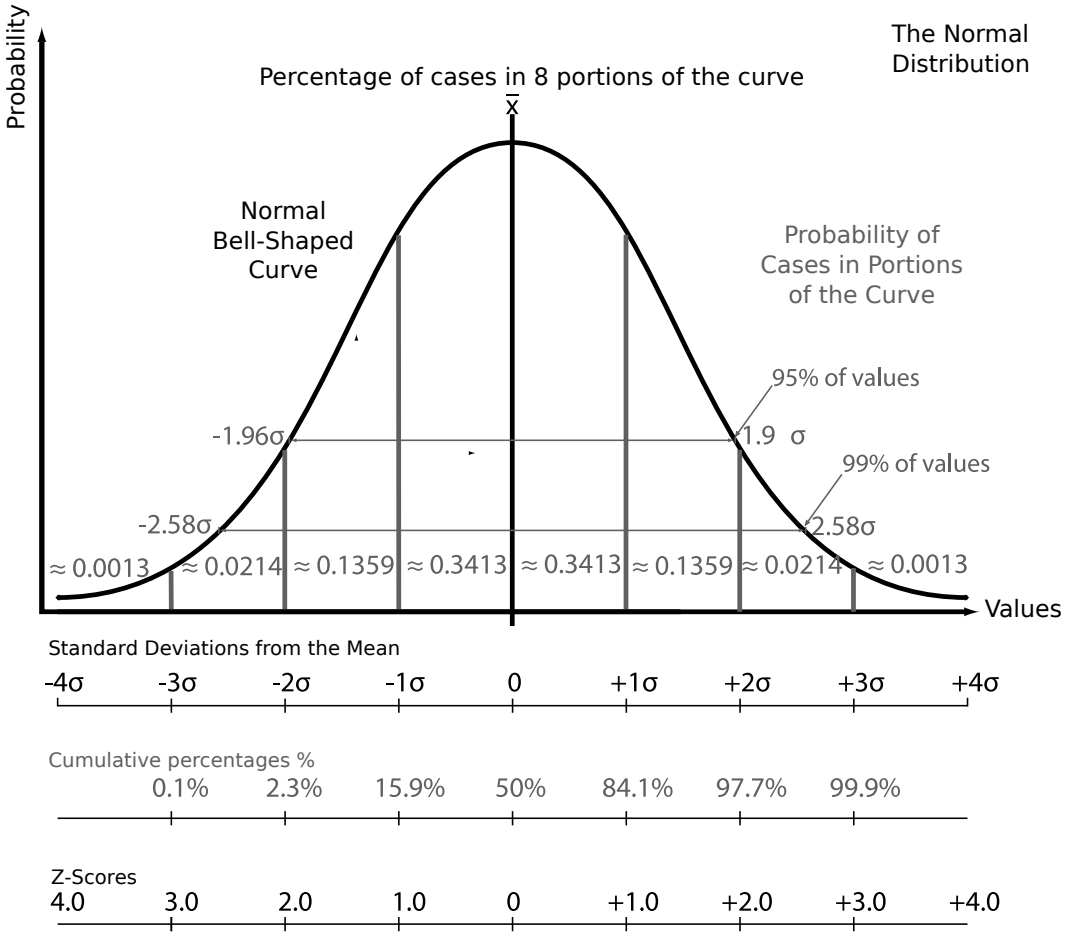

Figure 3.3: A sample demonstration of z-scores in a normal distribution

$$\text{where} \quad c = arg_{k=1,\ldots np}\,\text{find}(x_{(1:end,:)} = k) \tag{3.29}$$

where np is the number of operational regimes and c is cluster variable. This equation can further be extended into the following form.

$$\hat{x}_{(c)} = \sum_{i=1}^{np} \frac{x_c - \left(\frac{1}{n_c}\sum_{j=1}^{n_c} x_{cj}\right)}{\sqrt{\frac{1}{n_c - 1}\sum_{q=1}^{n_c}\left|x_{cq} - \mu_{cq}\right)\right|^2}} \tag{3.30}$$

$$c = arg_{k=1,\ldots np}\,\text{find}(x_{(1:end,:)} = k) \tag{3.31}$$

Common alternative approaches to the integration of multiple regression computing environments include robust regression (Ramasso, 2014a), Kalman

filter (Peng et al., 2012a) and principal component analysis (Liao and Sun, 2011; Zhao et al., 2011). Although these approaches can tackle the feature selection and health indicator estimation issues, their main shortcoming is that the proportionality of raw engine sensors may not be satisfied by the processed data if the dimensions of the operational conditions for the underlying dynamical system are large. Furthermore, if the time series of the raw engine data becomes noise-corrupted, the number of states and the complexity of these methods might rapidly increase, and therefore the data processing may incorrectly calculate initial wear levels. Multi-regime normalisation can accomplish the identification of data proportionality and wear level characteristics, but it can only process the standardisation when the training and test data are available at the same time. Thereby, the theoretical significance of this section is to provide multi-regime normalisation, as performed on raw data, in order to assign the health indicator as an output for the neural network filtering function. Data pre-processing is only used to transform the raw data into an understandable format for the automated filtering process of neural network training.

In Figure 3.4, the standardised sensors after multiple regime normalisation are shown for a single vector (ns=1). The regime differences are eliminated here, and the data is transformed into a form that allows for only one valid regime for all sensors. The process is applied to all regimes separately and the standardised sensors are reassembled to form the normalised dataset.

3.4.1.3 Health Indicator Assessment

The normalised data form a population which is homogeneous in terms of regimes but they are still multivariate in terms of number of sensors. Although all sensors can be regarded as working in a single operating condition and the dimensionality is reduced into a common scale, these multiple sensors

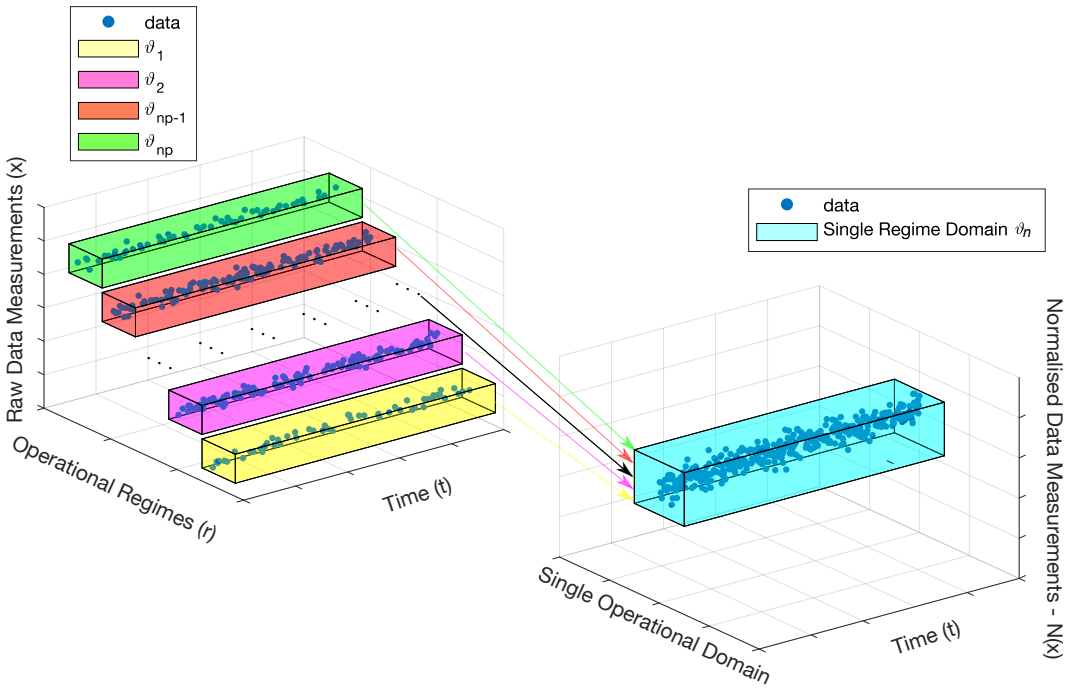

Figure 3.4: Normalisation of Multidimensional Data

may contain operational noise and perform poorly in terms of identifying the system's health degradation. This noise in the literature is commonly defined as Gaussian (Li et al., 2000; Menezes and Barreto, 2008; Lam et al., 2014) and its probability is given by the normal distribution (Cattin, 2013).

$$p_G(N(\varepsilon)) = \frac{1}{N(\sigma)\sqrt{2\pi}} e^{-\frac{(N(\varepsilon)-N(\mu))^2}{2N(\sigma)^2}} \tag{3.32}$$

where $N(\varepsilon)$ represent the normalised noise while $N(\mu)$ and $N(\sigma)$ represent the normalised mean value and the standard deviation.

In order to evaluate how well the sensors represent the degradation pattern, it is essential to eliminate those signals that do not adequately represent continuous normalised values exhibiting a monotonic exponential trend during the operational lifetime of the training units (Rigamonti et al., 2016). For this purpose, the prognostic parameter-choosing measures, "Monotonicity", "Prognosability", and "Trendability" (Coble, 2010), are used in this study to

assign the useful (and also meaningful) sensors to be used in further stages.

Suitability metrics and identification of prognostic parameters from data are used to characterise the lifetime of a system in its specific environment. Such an identification of suitable parameters is vital to accurate and precise RUL estimation. Three different parameter suitability metrics (proposed as monotonicity, prognosability, and trendability) are all important measure that should be considered in selecting the fitness of a parameter in RUL estimation (Coble and Hines, 2009). Therefore, these suitability measures are weighted to give the desired zero to one scale in which the measure outcome close to 1 demonstrates that the sensor is useful for RUL estimation, whereas an outcome close to 0 demonstrates that the sensor is a non-useful signal and not suitable for further prognostic consideration. As they are weighted into a common scale, the sum of their measures can determine the "fitness" of a candidate prognostic parameter.

$$fitness = monotonicity + prognosability + trendability \tag{3.33}$$

As the data is collected from a complex system, it is important to reduce the dimensionality of data to achieve accurate estimates of the derivatives.

Monotonicity is a straightforward measure used to understand whether a sensor always has the same underlying positive or negative trends which are necessary for prognosis. This measure characterises a sensor's general increasing or decreasing nature which indicates the degradation process in the system. It is defined by:

$$Monotonicity = mean\left(\left|\frac{\#pos\frac{d}{dx}}{n-1} - \frac{\#neg\frac{d}{dx}}{n-1}\right|\right) \tag{3.34}$$

where n is the number of trajectories in a particular history. The mono-

tonicity measure of a sensor population is computed by the average difference of the fraction of positive and negative derivatives for each trajectory. As the system degradation follows monotonic patterns in all trajectories in a dataset, this metric detects whether there is a characteristic underlying positive or negative trend of the sensor.

Prognosability is computed as the deviation of the failure threshold points for each trajectory divided by the average variation of the sensor during its entire operational cycles. The main purpose of this metric is to measure the spread of a sensor's failure value (or commonly known as threshold point) for a population of trajectories. If the failure points cluster in similar domains, the sensor is accepted as useful in terms of prognosability metric. The measure is exponentially weighted by the following equation to provide the desired 0 to 1 scale:

$$Prognosability = \exp\left(-\frac{\sigma_{failure}}{\mu_{failure} - \mu_{healthy}}\right) \tag{3.35}$$

Prognosability measures close to 1 suggest that the failure points in all trajectories are similar domains and the sensors are practical for prognosis, whereas the measures close to 0 indicates that the failure thresholds do not match with each other and the sensors are not capable of giving meaningful the prognostic calculations.

Trendability is calculated by the minimum absolute correlation computed among all the trajectories. The metric mainly detects whether the sensor for a population of trajectories have the same underlying trend pattern, and hence can be defined by the same parametric function. The mathematical expression for this is represented by:

$$Trendability = \min\left(|\text{corrcoeff}_{ij}|\right) \tag{3.36}$$

where corrcoeff$_{ij}$ is the correlation coefficients computed among all the training trajectories (n_{tr}), $i \neq j$ and $i, j = 1, \cdots, n_{tr}$.

According to the results of these prognostic parameter choosing measures, the useful sensors are selected for prognosis and employed in the health indicator assessment. However, it is necessary to emphasise the noise variations in the dataset and bring out strong patterns in the population. Assuming that the damage accumulation in data is monotonic and reveals itself in the system readings (Penna et al., 2012), a smoothing method can be set for the entire population. The noise and the random fluctuations that occur in a single trajectory can be removed or dissipated.

An often-used technique for such smoothing is that of the moving average (MA) method which reveals the underlying trend by taking the mean of a window size at each time index (Wei, 1994). Common alternatives to signal averaging for noise reduction in prognostics are the Z-Test (Daigle et al., 2010; Rigamonti et al., 2016), kernel smoothing (Wang et al., 2008; Wang, 2010) or polynomial fitting (Bektas and Jones, 2016). In these cases, the sensors are first fitted individually, and then the averaging is performed. By using such a means of processing, possible fluctuations in the sensors may not be dissipated and a closer estimate of the variability around the mean within the population might not be obtained. Also, similar to the modified moving average method, the principal component analysis can be used as a dimension-reduction tool to reduce a large set of sensors to a single HI that still contains data characteristics (Liao and Sun, 2011; Zhao et al., 2011). However, since the trajectories start at normal operating conditions and fail at a point with higher values, this approach can only be used in a unique trajectory rather than the entire dataset. Thereby, the assignment of health indicators (HI) may not be provided on a common scale.

For a random variable vector x made up of n scalar observations, the

moving average is defined by the following equation (Wei, 1994):

$$s_i = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.37}$$

where x is a random variable vector made up of n scalar observations.

In a similar way, the equation can be modified to represent the normalised condition monitoring data. The multivariate readings $x(1:n, 1:ns)$ are used to produce a single HI by taking the mean of all sensors at each time step.

$$s_i = \sum_{i=1}^{n} \frac{1}{ns} \sum_{j=1}^{ns} \hat{x}_{ji} \tag{3.38}$$

where n is the number of sensors instead of the observations.

3.4.2 Neural Network Filter

Since the multi-regime normalisation considers the overall population characteristics of data, ensuring integrity in the health index can be achieved with the population-based data standardisation. However, in the instance that new data is expected to be normalised, this process must be repeated and the entire algorithm might need re-arrangements. To avoid such a situation, a data-driven prognostic technique based on feed-forward neural networks with multiple regime normalisation is presented to calculate the asset performance levels with time. The multi-regime normalisation can be done once and then the health indicators can be used in the network training structure. The network function after training could filter any of the trajectories regardless of whether the data population characteristics change or otherwise.

The basic method for using neural networks is that data must be received from a representative population of cases and the output must be known by some well-confirmed method (Cross et al., 1995). According to this definition, the initially calculated HIs define the system outputs for a feed-forward

neural network model. The data filtering is designed to assume a role in dynamic modelling task by behaving as a normalisation and fitting system.

The ANN algorithm in this study uses the bias and weight objects of a network to store all the information that defines a neural network and the relation between input and output series. Then, these memory objects could describe filtered health indicators for any operational case applied into the trained network function.

3.4.2.1 Feed Forward Neural Network

In neural network filtering, ANN function takes the measurement of uncertainty bounds into account. Data from the multiple regimes are handled by a network mapping process between the raw values and the assigned HI. It is critical to choose an adaptive filtering method that can account for standardisation issues in addition to providing effective damage information.

This section offers a Bayesian regularised multilayer perceptron feed-forward neural network architecture that uses the defined HI outputs in the network training stage and generates a network function to fit the raw data trajectories into filtered health indicators. This stage includes arranging the network in such a manner that it is compatible with the multidimensional data case to be solved. The weights and biases are required to be tuned so that the network performance is optimised (Demuth et al., 2008). The governing equations for such a network function is denoted as:

$$\begin{aligned} y_{(t)} &= f\left[\mathbf{x}_{(t)}\right] \\ &= f_o\left\{b + \sum_{h=1}^{n_h} w_h f_h \left(b_h + \sum_{i=1}^{n} w_{ih} x_{(i)}\right)\right\} \end{aligned} \tag{3.39}$$

where the fixed real-valued weights, (w_i), are multiplied by the raw mul-

tidimensional condition monitoring data, (x_i), and the bias, b, is added. The neuron's output, y (which is calculated from the previous mode), is achieved as a result of the nodes and the transfer function of the neurons (Barad et al., 2012; Krenker et al., 2011).

In Figure 3.5, the multiple-layer neural network model of the methodology is shown. Each layer here is formed of its own weight matrix, bias vector, transfer functions, multidimensional condition monitoring data as input and calculated HI as output. The demonstrated structure is formed of a feed-forward model that takes a set of input vectors (raw data) as columns in a matrix, and then arranges another set of output vectors (HI) into a second matrix. Neurons, which are the building blocks of neural networks, evaluate these input state variables.

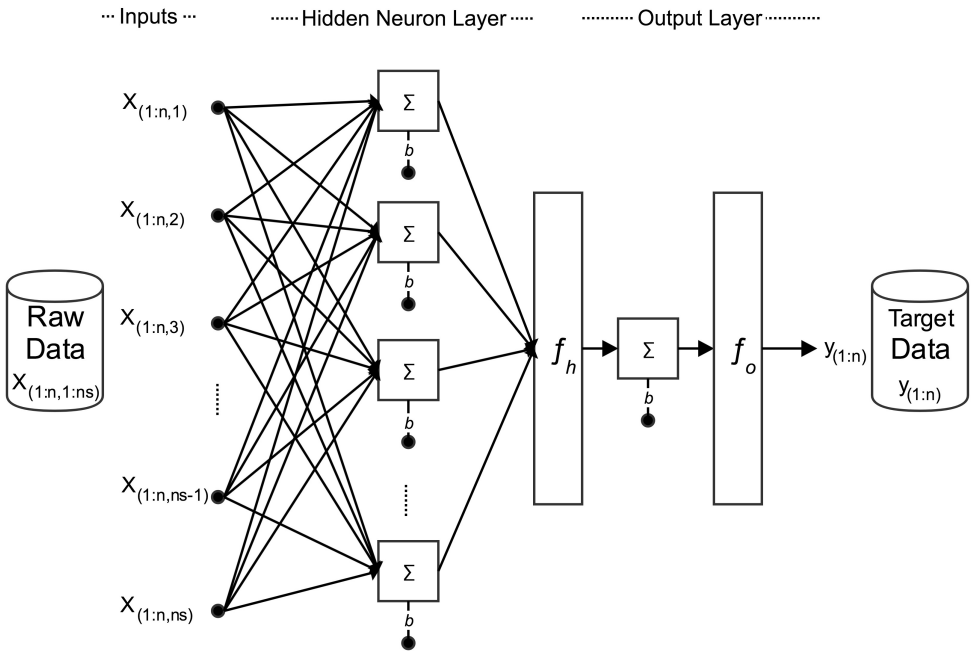

Figure 3.5: Feed-forward neural network with multidimensional input data and HI output

One major point in neural network training is that overheads might lead to poor network generalisation for the testing. To avoid such over-fitting, the

proposed feed-forward neural network model uses a common network training approach (Bayesian regularisation as discussed in section 2.3.3.3.3). This regularisation method typically needs more time in comparison to other methods due to the adaptive weight minimisation, but can result in satisfactory generalisations from difficult or noisy datasets (Demuth et al., 2008).

3.4.2.2 Network Library

While a network function trained with a particular data can provide the intended results, there may be the risk that if the network is trained with alternative inputs and outputs, the network results might come up with undesired estimations due to the changed weight and bias values. By considering this, alternative neural networks (f_{NN}) with different (but from similar domains) input and output values are trained for parallel problems. These networks form an artificial neural network library (L) for multiple calculations for each raw input.

$$L_{\{i\}} = f_{NN_{\{i\}}}, \quad i = 1, 2, \cdots, nl \tag{3.40}$$

where nl corresponds to the number of trained functions in the network library.

The proposed network library is a source of data filtering and normalisation, which implements multilayer feed-forward artificial neural networks in different operational cases with support for both fully trained network memory elements. The network HI generation for both training and test trajectories is maintained by integrating different knowledge sources to build a network, so that a collaborative estimation in different domains could be carried out to provide multiple results for each input data. The library includes a framework to allow for straightforward handling of raw data sets. It is adaptive to use this method with different trajectories, versatile for sensors, and compatible with the collaborative RUL calculations.

3.4.2.3 Filtered Health Indicators

Due to the reason that the library formed of multiple network functions provides multiple estimations for a single input, a final HI assignment similar to that in section 3.4.1.3 is necessary for reducing dimensionality and receiving a single target for RUL estimation. For the following stages of similarity based multi step ahead prediction, the moving average of all these library estimations (referring to equation 3.37 and equation 3.38) is carried out to filter the final HI.

$$s_r = \frac{1}{p} \sum_{q=r}^{r+p} \frac{1}{nl} \sum_{j=1}^{nl} a_{jq} \tag{3.41}$$

where the sequence s_r is the total mean of p-moving average of nl number of HIs. The window size (p-moving average) is defined as a numeric duration scalar for the upcoming time instants and the average contains the parameters in the current location as well as the p number of following neighbours. The window is also widened prior to the s_r and the centred moving average equation is denoted as:

$$s_r = \frac{1}{2p+1} \sum_{q=r-p}^{r+p} \frac{1}{nl} \sum_{j=1}^{nl} a_{jq} \tag{3.42}$$

For an example of a two-point average, the matrix with a window size of 2 and n HIs is defined as:

$$\begin{bmatrix} s_{r-2,1} & s_{r-2,2} & \cdots & s_{r-2,nl} \\ s_{r-1,1} & s_{r-1,2} & \cdots & s_{r-1,nl} \\ s_{r,1} & s_{r,2} & \cdots & s_{r,nl} \\ s_{r+1,1} & s_{r+1,2} & \cdots & s_{r+1,nl} \\ s_{r+2,1} & s_{r+2,2} & \cdots & s_{r+2,nl} \end{bmatrix} \tag{3.43}$$

or, for a three-point average, a matrix with a windows size of 3 is as:

$$\begin{bmatrix} s_{r-3,1} & s_{r-3,2} & \cdots & s_{r-3,nl} \\ s_{r-2,1} & s_{r-2,2} & \cdots & s_{r-2,nl} \\ s_{r-1,1} & s_{r-1,2} & \cdots & s_{r-1,nl} \\ s_{r,1} & s_{r,2} & \cdots & s_{r,nl} \\ s_{r+1,1} & s_{r+1,2} & \cdots & s_{r+1,nl} \\ s_{r+2,1} & s_{r+2,2} & \cdots & s_{r+2,nl} \\ s_{r+3,1} & s_{r+3,2} & \cdots & s_{r+3,nl} \end{bmatrix} \tag{3.44}$$

This is a directional window formed by a temporary duration matrix. Its size depends on two different elements, the number of library estimations "nl" and the scalars of moving average windows size "p". Considering both posterior and prior values, as well as the sequence itself, the exact size of the window is $(2p+1) \times nl$. The calculation includes nl elements in the current location, $p \times nl$ elements before the current location, and $p \times nl$ elements after the current location.

Regarding the initial starting and the final ending points, this moving average method for a full matrix of library estimations is modified as:

$$s_i = \begin{cases} s_i = \frac{1}{p+i} \sum_{q=1}^{p+i} \frac{1}{nl} \sum_{j=1}^{nl} a_{jq} & \text{if } i-p<0 \\ s_i = \frac{1}{p+(l-i+1)} \sum_{q=i}^{p+(l-i+1)} \frac{1}{nl} \sum_{j=1}^{nl} a_{jq} & \text{if } i+p>l \\ s_i = \frac{1}{2p+1} \sum_{q=r-p}^{r+p} \frac{1}{nl} \sum_{j=1}^{nl} a_{jq} & \text{if } i-p>=0 \\ & \text{and if } i+p<=l \end{cases} \tag{3.45}$$

where l is the length of the operational case. The insufficient dimensions ($i-p<0$ and $i+p>l$) are defined either as a positive scalar for the starting

points or a scalar smaller than the length of matrix for the ending points. The rest have can have full dimensions of the moving average and operate along with the direction of the specified window move. Accordingly, the modified moving average method is able to filter both ending and starting points of time series.

3.4.3 Multi Step Ahead Prediction

In the future failure mode, the filtered health indicators are stored to calculate the pairwise distance relations for the life time predictions. A similarity-based RUL estimation model identifies the best matching training units for each test case and makes future multi-step predictions over filtered health indicators. A particular issue encountered when attempting to make meaningful long-term predictions is that of taking account of different kinds of uncertainties arising from various sources.

3.4.3.1 Pairwise Distance Relation

The similarity-based prediction algorithm proposed in this research estimates the future behaviour of the systems only when sufficient training data to map out the damage space is present and has been examined by comparison with a robust standardisation. The filtered health indicator derived from the data must give a realistic representation of system performance and manage entire trajectories correctly. For example, when more information about historical damage propagation becomes available, the filtering should be devised to narrow the errors in identification of trajectory characteristics, and the results should demonstrate the relationships between the historical performance deterioration of initially trained subsets and RUL predictions of test subsets.

RUL prediction of test data is calculated through the best-matching

training units that have a run-to-failure history (Wang et al., 2008; Lam et al., 2014; Ramasso, 2014a; Eker et al., 2014; Peng et al., 2012b). Distance matrices are used in similarity-based estimation for pairwise distance estimations. These distances are then used to measure the similarity between training and test HIs.

Each training curve is accepted as a baseline for the degradation pattern for prediction. The pairwise distance between the pairs of training and test HIs can be regarded an error rate for each corresponding point. According to this error rate, the similarity between the degradation trajectories of two diverse instances can be calculated first; then, the failure threshold point of the test trajectory is estimated based on the actual failure point of the corresponding training trajectory. Finally, the RULs estimated from multiple training data can be fused to compute a final RUL estimation.

Euclidean distance is used to measure the ordinary distance between the points of two corresponding trajectories in Euclidean space. Although various distance-based methods can be used in the model, since the trajectories are in a continuous space where all dimensions are properly scaled and relevant, the Euclidean is an appropriate choice for the distance function. With this method, Euclidean space becomes a metric space and calculates the distance between pairs of objects in a data matrix that represents the similarity of corresponding parts.

The Euclidean distance between the vector of the test (p) and corresponding part of the training trajectory (q) is the length of the line segment connecting both vectors ($\bar{pq}$). In Cartesian coordinates, when "$p = (p_1, p_2, \cdots, p_n)$" and "$q = (q_1, q_2, \cdots, q_n)$" are two vector points in Euclidean n-space, the distance (d) between p and q, or the reverse, is calculated by Pythagoras' theorem, which can be written as an equation relating the lengths of the sides p, q and d (Sally, 2007).

$$\begin{aligned} \mathrm{d(p,q)} = \mathrm{d(q,p)} &= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}. \end{aligned} \tag{3.46}$$

Figure 3.6 shows pairwise distance calculations between a particular test sample and two sets of training sample observations which include run to failure degradation progress. As seen in this example, the best-matching training units for the test data can be located in the ongoing parts of the curve rather than being in the initial stages. The testing curve is moved in a step-by-step manner throughout the base curve to identify the minimum pairwise distance between the trajectories. This illustrates the fact that the relation between the test and training samples is calculated and stored at each step in order to find the possible best-matching part of the training domain. Given the measure of the distance between each pair, the matching locations are used as feedback to complete the missing parts of the test trajectories, and the algorithm continues to the next training time units by repeating the same process. The pairwise distance over each time unit is given by the following equation:

$$d(te, tr, j) = \sum_{j=n_{te}}^{n_{tr}} \sqrt{\sum_{i=1}^{n_{te}} (te_i - tr_{i+j})^2}. \tag{3.47}$$

where n_{te} is the length of test trajectory and n_{tr} is the length of training trajectory (the base curve).

3.4.3.2 RUL Estimation

Once the testing curve has been moved throughout the baseline, the minimum of the stored pairwise distance values is identified as the best-matching part

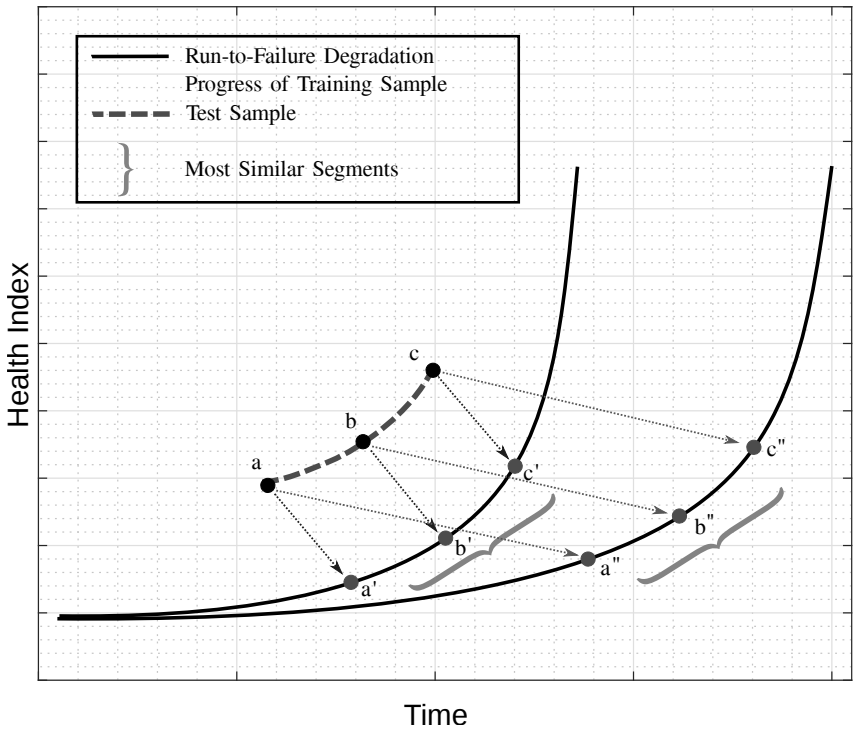

Figure 3.6: Similarity Methodology

between the test data and the training baseline, $Mn_{(te,tr)}$.

$$Mn_{(te,tr)} = min\left(d_{(te,tr,1)}, d_{(te,tr,2)}, \cdots, d_{(te,tr,n_{tr}-n_{te})}\right) \tag{3.48}$$

The location of the best-matching time instant at the training baseline, $L_{te,tr}$, is determined by:

$$L_{te,tr} = \arg\text{find}\left(d_{(te,tr,j)} = Mn_{(te,tr)}\right); j = 1, 2, \cdots (n_{tr} - nte) \tag{3.49}$$

A presentation of minimum distance and the estimated RUL are shown in Figure 3.7. The baseline data before this location is accepted as non-useful information to be removed, and the remaining part is used as a representation of the test trajectory's future behaviour.

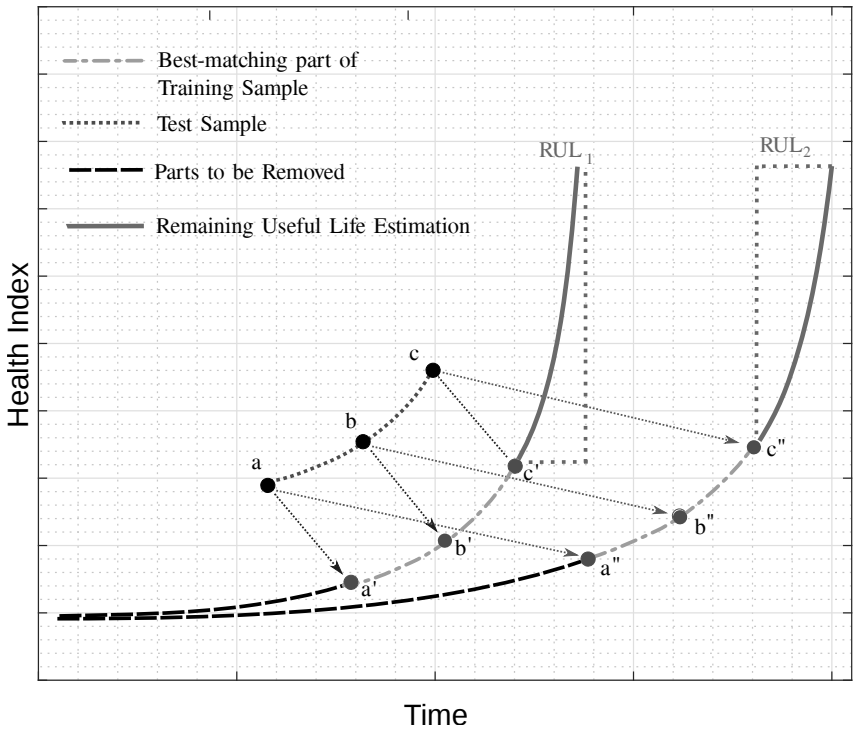

Figure 3.7: RUL Estimation

By using the calculated location of the minimum Euclidean distance, a RUL matrix for each test trajectory at each training baseline can be calculated by the following formula.:

$$RUL_{te,tr} = n_{tr} - (n_{te} + L_{te,tr}) \tag{3.50}$$

The length of time series after the identified location of best matching time instant, $L_{te,tr}$, gives the estimated RUL of the test trajectory. However, a single RUL prediction with a single training baseline can be extended to a collaborative group of best matching training trajectories and the estimations with more training samples can increase prediction performance in multi-step ahead RUL estimations. In other words, the data collaboration effort can be performed from multiple suppliers with a wide variety of degradation cases rather than using separate and self-contained sources.

3.5 Summary

This chapter presents a prognostic framework which combines three advanced approaches: multi regime normalisation, feed-forward neural network and similarity-based prediction. These approaches are proposed to be implemented on multidimensional condition monitoring data to evaluate RUL of complex systems operating under different conditions.

First contribution in the methodology is reducing the dimensionality of condition monitoring data by using a health indicator filtering method of feature extraction which can standardise the entire data population with regard to the characteristics of individual operational cases. This is a hybrid method which uses multi regime normalisation to assign target outputs for feed forward neural network and can filter the HI's of different operational cases with regard to each other or any potential novel cases. Consequently, the proposed methodology can preserve operational features when applied to actual condition monitoring data, exhibiting health degradation.

Another contribution is in demonstrating the strengths of collaboration in RUL estimation. In practice, a single dimension reduction method can be applied to extract the operational features or the above mentioned multi regime normalisation and neural network approaches can be used individually to filter health indicators. However, such an individual application would have drawbacks on the future failure mode. Since the hybrid method can preserve operational features regarding to existing operational cases in a data or the novel ones, the RUL prediction can be made through the similarities between different existing or novel samples which can be standardised into a common scale. Therefore, the similarity-based prediction that receives the filtered health indicators from the hybrid feature extraction model is promising to be applied in complex system prognostics, where the regimes occurring in degradation signals are multidimensional.

In terms of practicability, the proposed methodology will beconsidered in a complex system domain and the framework will be applied into multidimensional data. In the next chapter, these scenarios will be discussed and a case study will be introduced.

Chapter 4

Case Studies

In Chapter 3, a prognostic framework has been demonstrated for dealing with prognostic prediction problem in complex systems. In this section, the benefits of using the proposed methodology will be demonstrated in a case study, where it can account for life cycle uncertainties in degradation to deal with challenges associated with complex systems, including multidimensional regimes occurring in degradation signals.

The discussion of major points of the prognostic algorithm will be presented and their subsequent extensions will be described with a particular focus on their developments in the field of complex systems. As in section 4.1, an application of the proposed methodology in gas turbine engine will be defined as well as a further discussion on why gas turbines are an appropriate case study for the model and how they exemplify a complex system and research objectives.

A discussion is provided as to the nature of the gas turbine data required to run the model both in terms of training data and test data. Subsequently, a critical discussion of the methodology and its application to alternative cases is laid out.

4.1 Application: Gas Turbine Engine Case

Gas turbine engines are widely used in propulsion systems (particularly in aircraft propulsion) and in power generating systems (Meier and Pettit, 1989). Their monopoly is connected with their ability on high power, thrust and efficiency. However, gas turbine engines are very complex and sensitive systems which have to operate safely and efficiently over a range of extreme operating conditions (Patel et al., 1996). Due to this necessity, engine manufacturers (e.g. General Electric, Pratt & Whitney and Rolls-Royce) have performance-based plans in which their compensation is directly related with the product availability (Marinai et al., 2004; Kim et al., 2007). This relation gives priority to the health monitoring applications of gas turbine engines to reduce life-cycle costs, improve engine reliability as well as availability (Marinai et al., 2004; Li and Nilkitsaranont, 2009).

Engine health monitoring has been in existence for decades and technological advancements have improved the availability of the monitored measurements and the use of data (Mazdiyasni, 2008). Using these condition monitoring measurements for engine degradation forecasting or engine prognostics is very challenging because of the great uncertainty associated with gas turbine complexity and operating conditions (Li and Nilkitsaranont, 2009).

Such engineering systems operate under different operational margins and are composed of various subsystems, each of which is embodied by a particular set of sub-subsystems or components (Zaidan, 2014). The engine behaviour is not one dimensional due to the flight changes in recurrent phases: taxi, take-off, climb, cruise, descent, etc. (Lacaille and Bellas, 2014). The measured sensors in condition monitoring have accordingly superimposed operational margins at any given time instant. Therefore, an application of the methodology in gas turbine engine prognostics is considered as an appropriate case study for complex systems operating under different operational condi-

tions and comprising multiple interacting subsystems.

Aircraft engine condition monitoring provides early warning of ongoing or impending failures, and allows the operators to initiate preventative maintenance tasks to reduce reliability risks, unscheduled delays and other more serious system failures (Tumer and Bajwa, 1999). The main objective in such maintenance tasks is to use signal information as a reliable indicator of growing faults and impending potential failures that may be noticed and repaired during inspection; however, engine condition monitoring requires the man-in-the-loop to analyse the data and correlate with other information sources (Volponi, 2014). Therefore, they overly depend on the ground power plant analyst, and cannot define the performance level simultaneously. Moreover, the majority of fielded systems are not properly instrumented for relevant data collection and/or are highly restricted in their ability to share such data due to proprietary and confidentiality constraints (Ramasso and Saxena, 2014). For these reasons, the nature of the prognostics problem has its own challenges in terms of dataset availability and monitoring data (Eker et al., 2012)

In the absence of primary operating engine data, it is necessary to use data sources based on gas turbine performance simulations (Kurzke, 2017; GPA, 2013; Visser and Broomhead, 2000; MentorGraphics, 2013; Liu et al., 2013a; Frederick et al., 2007) and/or various other sources such as component models, maintenance histories and sensor measurements (Volponi, 2005). Although these gas turbine simulations can provide sufficient information for prognostic algorithm development, there is still the requirement to collect common datasets and a mutual comparison to validate the methods introduced by different researchers. In order to perform prognostic prediction effectively, the researchers are expected to develop their methods using common datasets of a minimum sample size (Eker et al., 2012). The lack of shared data sets to validate the methods has been the greatest obstacle to improved progress in

the prognostic literature (Ramasso and Saxena, 2014). To overcome this issue, several simulations have been introduced and, most notably, a variety of prognostic datasets have been published through the NASA Prognostics Center of Excellence (PCoE) data repository (PCoE, 2014). These publicly available datasets have been used by many researchers to develop and compare their algorithms. The most applied and used datasets in these are the "Turbofan Engine Degradation Simulation Data Set" and "PHM08 Challenge Data Set". Both are gas turbine simulations and are modelled in a similar way to each other. Their high application rates have resulted in the introduction of a significant number of methods in the literature. Besides, the score leader board provided by PCoE (2008) is an effective method for validation with existing methods in the literature.

4.1.1 C-MAPSS Simulation and Datasets

The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) is a NASA developed tool used to simulate a realistic large commercial turbofan engine. The software was coded in MATLAB (The MathWorks, Inc.) and Simulink (The MathWorks, Inc.) environments with a number of editable input parameters that allow the users to enter values specific to their own applications regarding environmental conditions, operational profile, etc. (Frederick et al., 2007).

C-MAPSS programme was used to implement the simulation of the PHM08 Challenge Data Set and Turbofan Engine Degradation Simulations (Saxena and Goebel, 2008a,b). All data sets differ from each other and were simulated under various combinations of regimes, operational conditions and fault modes. It is recorded that several sensor channels characterise fault evolution during operation. The data sets are made publicly available for the model training and the validation of results (Saxena and Goebel, 2008b).

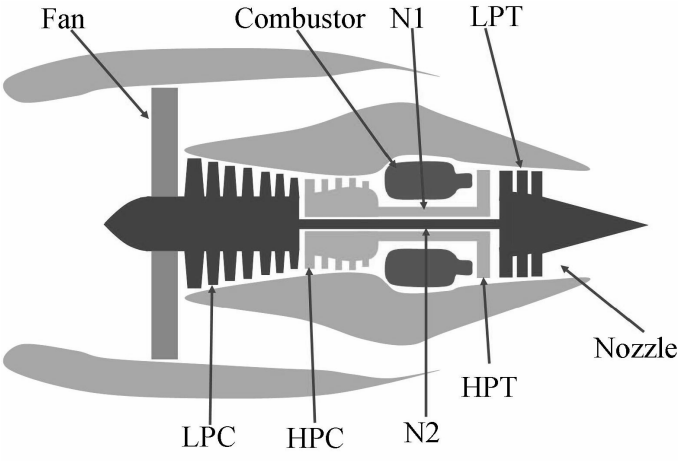

Figure 4.1: A simplified diagram of an engine simulation modelled in C-MAPSS (Frederick et al., 2007)

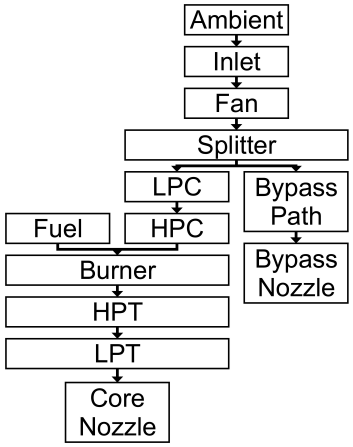

Figure 4.2: Subroutines of a model (Frederick et al., 2007; Saxena et al., 2008b)

The engine diagram in Figure 4.1 demonstrates the fundamental parts of an engine simulation, and the flow chart in Figure 4.2 demonstrates how the different subroutines are assembled in the simulation. The comprehensive control systems illustrated in these figures are formed of the following fragments (Frederick et al., 2007):

- A fan-speed controller for the specification of throttle-resolver angle
- Three high-limit regulators to avoid the engine from exceeding its own design limits of engine-pressure ratio, core speed, and high-pressure turbine exit temperature
- Four limiting regulators to avoid static pressure at the high-pressure compressor exit from going off too low
- Core speed acceleration and deceleration limiters
- A comprehensive logic structure integrating the control-system fragments in a similar manner to real engine controllers
- A power-management system allowing engine operation over a wide range of thrust levels covering the full range of flight conditions.

- In addition to the engine model, an atmospheric model is included with the capability of operation at

  – altitudes from sea level to 40,000 ft
  – Mach numbers from 0 to 0.90
  – temperatures from 60 to 103 °F

C-MAPSS aero engine degradation simulations possess the following characteristic features that make them both convenient and suitable for the development of prognostic algorithms on multi step ahead RUL estimations (Ramasso and Saxena, 2014; Saxena and Goebel, 2008a,b):

- Each data set contains multivariate and multidimensional time series representing sensor magnitudes over time and three operational settings that have a significant effect on engine performance and variations within operational regimes. Therefore, the data sets can closely imitate real systems by exemplifying multidimensional operations of complex non-linear systems.

- Data sets are divided into training and test trajectories, the latter being individual subsets. In the training trajectories, operational cases of complete run-to-failure data are formed, which are supposed to be used to train multi step ahead life prediction algorithms. The test trajectories, on the other hand, can only set up via shorter instances with data up to a certain time prior to adopting system failure.

- The sensors are contaminated with operational regimes and noise to simulate instability within parameter readings during operation. Also, each trajectory is assigned a distinct degree of initial wear and manufacturing variation, which is considered normal and unknown to the user.

-The fault effects are hidden on account of operational conditions and regimes, which is yet another common feature of most real-life operational systems.

-Raw values at each time point in data are regarded as a snapshot of the parameters taken during a single cycle, and each column corresponds to a different variable (see Table 4.1).

Table 4.1: PHM08 challenge data set parameters available to participants as sensor data Saxena et al. (2008b)

Parameters	Symbol	Description	Unit
Unit	—	—	—
Time	—	—	t
Setting 1	—	Altitude	ft
Setting 2	—	Mach Number	M
Setting 3	—	Sea-level Temperature	°F
Sensor 1	T2	Total temperature at fan inlet	°R
Sensor 2	T24	Total temperature at LPC outlet	°R
Sensor 3	T30	Total temperature at HPC outlet	°R
Sensor 4	T50	Total temperature at LPT outlet	°R
Sensor 5	P2	Pressure at fan inlet	psia
Sensor 6	P15	Total pressure in bypass-duct	psia
Sensor 7	P30	Total pressure at HPC outlet	psia
Sensor 8	Nf	Physical fan speed	rpm
Sensor 9	Nc	Physical core speed	rpm
Sensor 10	epr	Engine pressure ratio	—
Sensor 11	Ps30	Static pressure at HPC outlet	psia
Sensor 12	phi	Ratio of fuel flow to Ps30	pps/psi
Sensor 13	NRf	Corrected fan speed	rpm
Sensor 14	NRc	Corrected core speed	rpm
Sensor 15	BPR	Bypass Ratio	—
Sensor 16	farB	Burner fuel-air ratio	—
Sensor 17	htBleed	Bleed Enthalpy	—
Sensor 18	Nf_dmd	Demanded fan speed	rpm
Sensor 19	PCNfR_dmd	Demanded corrected fan speed	rpm
Sensor 20	W31	HPT coolant bleed	lbm/s
Sensor 21	W32	LPT coolant bleed	lbm/s

LPC/HPC=Low/High Pressure Compressor - LPT/HPT= Low/High Pressure Turbine

Each trajectory in a dataset is from a different operational instance of a complex aero engine system under dynamic operating regimes. The data

sets can be regarded as a fleet of the same type of aircraft. Since data can be collected from numerous samples, it is possible for algorithms to extract the behaviours of each trajectory and make collaborative calculations for the different courses of system actions.

Table 4.2: Number of trajectories and regimes in C-MAPSS data sets

Dataset:	Training Trajectories	Test Trajectories	Regimes
FD001	100	100	1 (sea level)
FD002	260	259	6
FD003	100	100	1
FD004	248	249	1 (sea level)
PHM08 Test	218	218	6
PHM08 Final Test		435	6

Table 4.2 summarises the fundamental differences between datasets. In all subsets, the test sets have a corresponding training set. FD002 and FD004 are formed of six different operational regimes, while FD001 and FD003 include only one operational condition. FD003 and FD004 also include an extra fault mode (fan degradation). Since each data set has a particular number of operational conditions and fault modes that can have a direct impact on their performance, only the trajectories in the same data set can be regarded being from an identical system. On the other hand, PHM08 Challenge Data Set only includes one corresponding training set for two different test sets. The developers are expected to model their algorithms using this single set of training data for both 'test' and 'final test' sets provided in the same package. Also, unlike the Turbofan Engine Degradation Simulation, the true RUL values are not provided for the challenge participants. Instead, users are asked to upload their "test" results on the repository page to receive their scoring function results (Saxena and Goebel, 2008a). Since the true RULs are not given, it is not

possible to validate the results with any other prognostic metrics. For the "final test", the users can only send their results once, and therefore this subset does not allow a test-and-trial type of submission and can only be validated by a third party.

The validation uses the taxonomy of RUL performance measures and prognostic metrics presented in the Literature chapter (section 2.4). In Appendix A, a comparison of the well-known publications is provided. The metric results obtained by particular works are summarised in table A.1 and the leader board for the PHM08 data challenge is presented in table A.2.

4.2 Preparing Multidimensional Data

The performance of a gas turbine engine degrades over time. To identify such performance degradations, it is necessary to apply a prognostic data analysis that uses historical health information provided by the engine parameters (Li and Nilkitsaranont, 2009).

Figure 4.3 shows a flowchart representing data processing starting from the initial raw turbofan engine data to multi-step ahead RUL estimations. These instructions describe an application of the methodology that, when executed properly, can ensure a case study for gas turbine engine data.

For multiple regression aircraft engine data, different flight operational conditions are comprised of a range of values for operational conditions such as altitude, Mach number or throttle resolver (Saxena et al., 2008b). These operational condition margins change as system degradation takes place, and each includes several noise factors, characteristic initial wear levels and lack of information on the operational regime effects. The application, thereby, needs to deal with the uncertainty problem at various levels of the prognostic modelling.

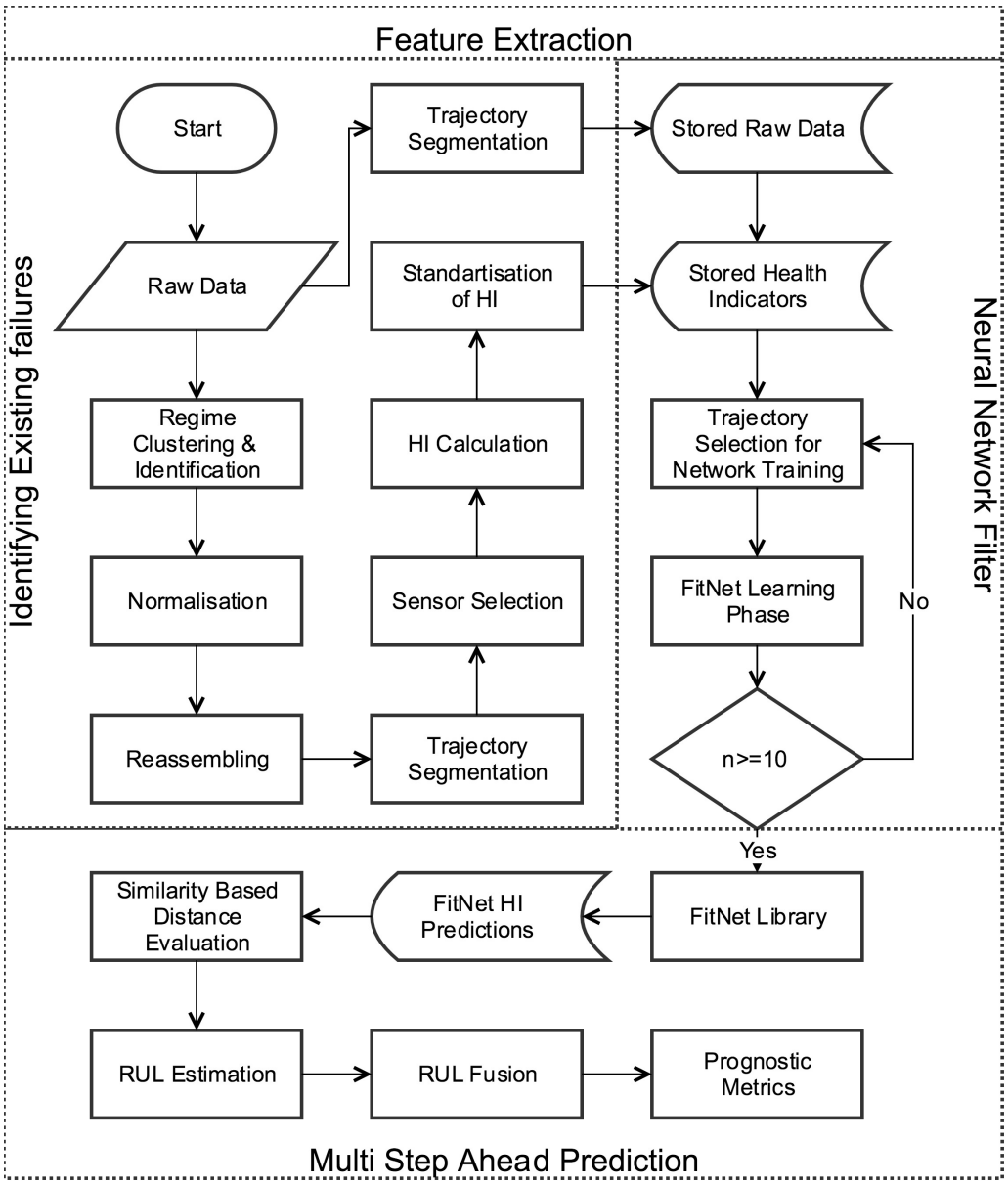

Figure 4.3: Flowchart representing data processing and multistep ahead remaining useful life calculation

Figure 4.4 provides the characteristic features of raw sensor values in a sample training engine trajectory (from PHM08 Challenge Data Set). As it is shown in this figure, raw data is demonstrated as it is collected from an engine source and has not been subject to any processing or cleaning methods. Also,

the data has not been manipulated by any other algorithm, and the sensor values are not standardised or regulated.

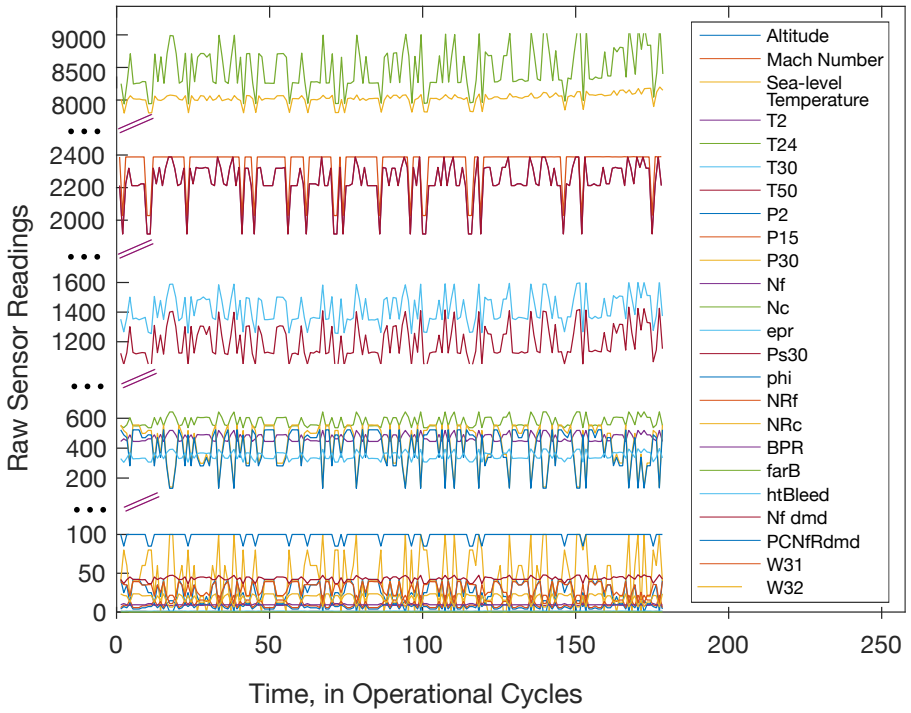

Figure 4.4: Raw Data Characteristics

The range of raw engine sensors at their maximum and minimum values is very wide and requires an organisation to reduce data redundancy and improve data integrity. For example, a given sensor may have a range of 8000 to 9000 while the range of others may span many orders of sensor magnitudes. Regardless of their range, these sensors represent complicated system conditions, and some of them are directly related to the system degradation (Saxena and Goebel, 2008a,b). Further observations are applied to check data integrity across the multiple sensors and to identify the regimes and degradation patterns in sensor measurements.

Figure 4.5 represents the processing behaviour of a single raw sensor (sensor 11 - Static pressure at HPC outlet). The plot illustrates that the data is highly scattered, multidimensional and it is difficult to plot a line of best

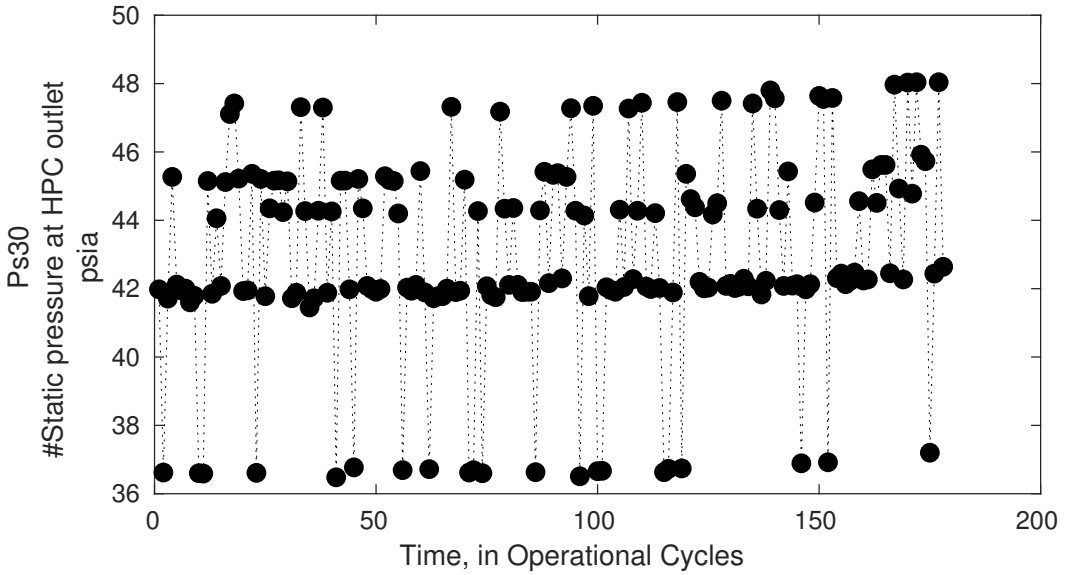

Figure 4.5: Sensor 11 - Static pressure at HPC outlet

fit representing the system degradation. A meaningful observation and/or understanding from the raw sensor is not possible without the identification of the degradation pattern from the multidimensional and noisy sensors. In pursuance of achieving a useful HI for target achievement in neural network training, a complex data processing approach is needed for feature extraction and selection. The characteristic features of multidimensional raw data under multiple system conditions need to be extracted, and then the useless and misleading information caused by the noise during operation need to be removed.

From the observations obtained from the single raw sensor monitoring shown in Figure 4.5, it is noticed that certain sets of data points cluster in similar domains than those in other regimes. When the plot is set by certain axis limits and aspect ratios, an exponential behaviour indicative of the monotonic system degradation can be observed in the plot. In Figure 4.6, the axis index for Sensor 11 is limited so that the "y-axis" ranges from 41.6 to 42.8 instead of 36 to 50, as in Figure 4.5. The raw sensor values in this range (or domain) can

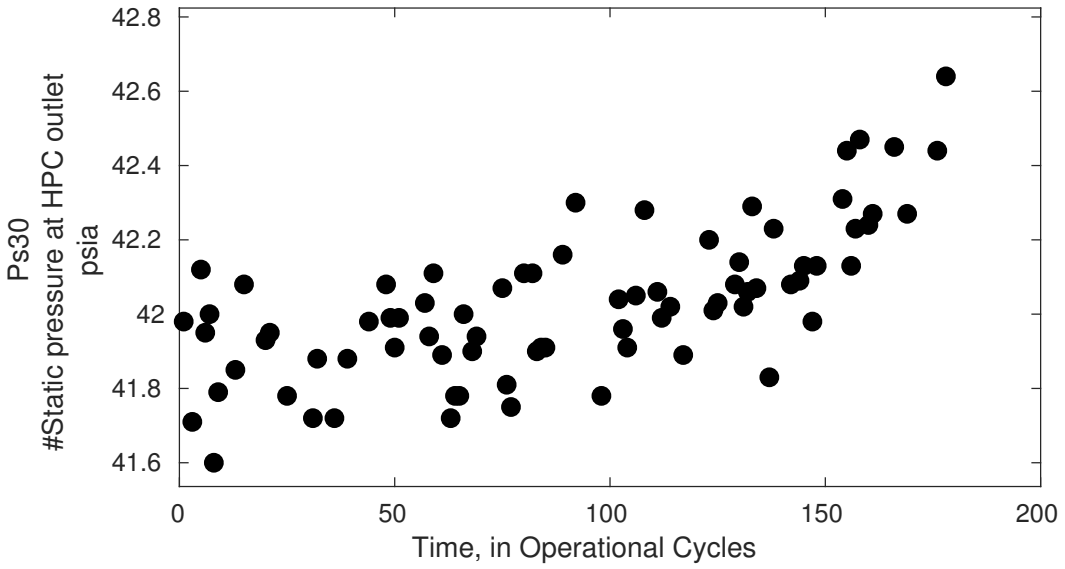

Figure 4.6: Sensor 11 - Static pressure at HPC outlet (y-axis limited)

be regarded as lying within a certain operational regime in which understandable prognostic behaviour is available for identifying the engine health levels. This behaviour follows a predictable/sequential degradation pattern similar to the aircraft engine damage propagation model of Saxena et al. (2008b). When the selected groups of data points are monitored in various trajectories to assess their response, it can be concluded that the response is uniform over the entire range of trajectories. However, the number and magnitude of axis limits and aspect ratios differ at different sensors.

On the basis of the using axis-related evidence as a starting point for further investigation, the identification of the data alignment characteristics is proposed as an explanation for the degradation phenomenon. The first step of data processing is, therefore, to classify these particular regimes in all trajectories.

The operational settings given in the engine data have a direct impact on the sensors, and the multiple regimes can be found by finding the optimum number of clusters in these settings. However, clustering cannot be regarded

as one specific method to the operational settings, but a general task to be resolved.

It is unlikely that all operational settings in different data will respond in similar manners to a clustering method. Some will respond better than others, and some may fail to respond at all. For example, different data sets of the C-MAPSS Turbofan Engine Degradation Simulation have different characteristics in operational settings and a clustering method developed on a single dataset may not work on others without a further modification of its parameters.

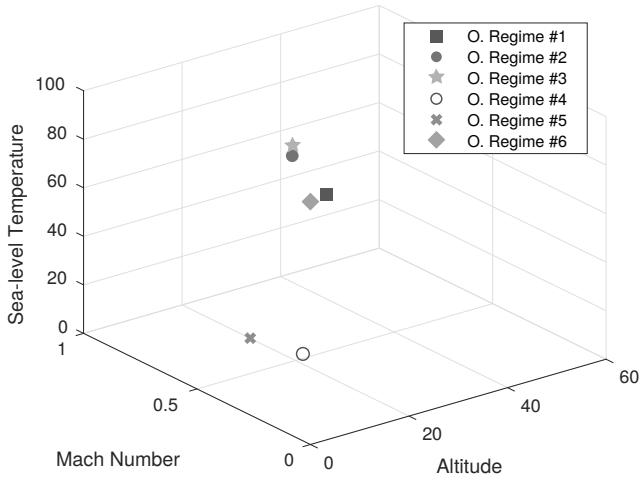

Figure 4.7: Operational settings in different regimes

The clustering methods can be obtained in alternative ways that most notably differ in their concept of what actually establishes a cluster and how to effectively detect them (as like in Figure 4.7). For complex cases, as in the PHM08 challenge dataset, the common notion of multiple regime clustering can be arbitrary, such as k-means (Lam et al., 2014), fuzzy c-means (Sugeno et al., 1993), Gaussian mixture models (McLachlan and Peel, 2004), nearest-neighbour clustering (Ramasso, 2014a) or neural network clustering (Ultsch, 1993) etc. Although these methodologies appear different, their result in the C-MAPSS dataset match with each other. This is mainly because the centres of

three operational settings (altitude, Mach Number and sea-level temperature) indicates the regimes in data as can be seen in Figure 4.7.

Clustering can also be applied by using particular sensors that have unique and constant values within each regime. Sensor 1, shown in Figure 4.8, is formed of six different fixed values in the course of all trajectories. It is accepted that the quantity of regimes in the dataset corresponds to the unique values in these fixed sensors.

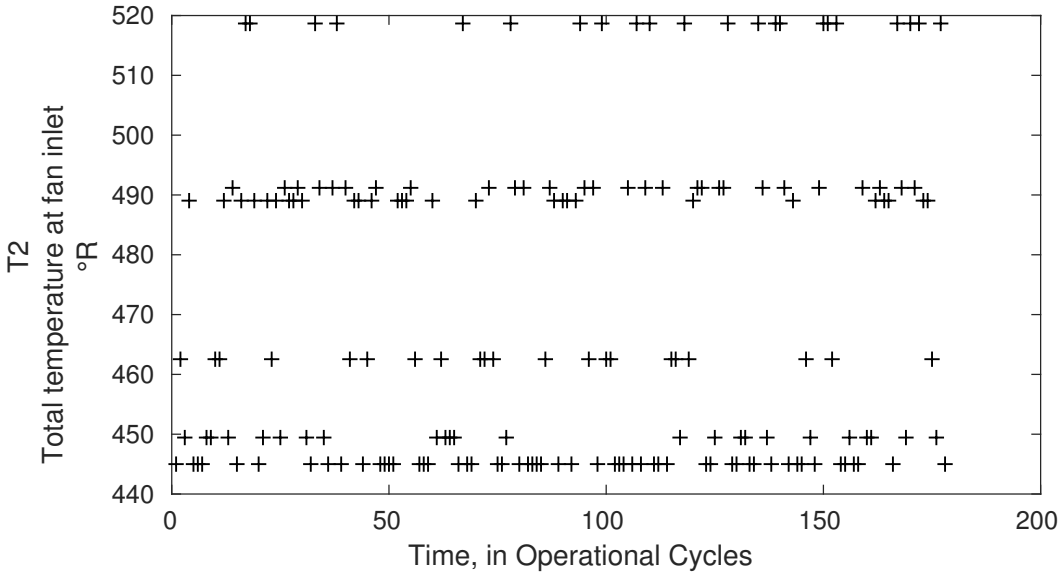

Figure 4.8: Sensor 1 with constant values at each regime domain

Table 4.3: Corresponding values for regimes

Regime	Operational Setting 1	Operational Setting 2	Operational Setting 3	Sensor 1
#1	10.0047	0.2501	20	489.05
#2	0.0015	0.0003	100	518.67
#3	34.9986	0.8401	60	449.44
#4	20.0031	0.7005	0	491.19
#5	42.0041	0.8405	40	445
#6	25.0051	0.62	80	462.54

Table 4.3 represents the consistency in the corresponding values for all regimes. Unlike for Sensor 1, operational settings 1 and 2 have slightly different values at each regime recurrence, but they are nevertheless gathered on certain boundaries. Operational setting 3, on the other hand, has six fixed values for PHM08 datasets but it differs in other C-MAPSS datasets and includes fewer numbers.

In this research, instead of developing a complex clustering method, a simple clustering function returns the unique values for "Sensor 1" for regime assignment. All sensors are classified in such a way that the sensors in the same row with the assigned unique values operate in the same regime. It is observed that this clustering function is valid for all C-MAPSS datasets and can be used as a common method across different applications.

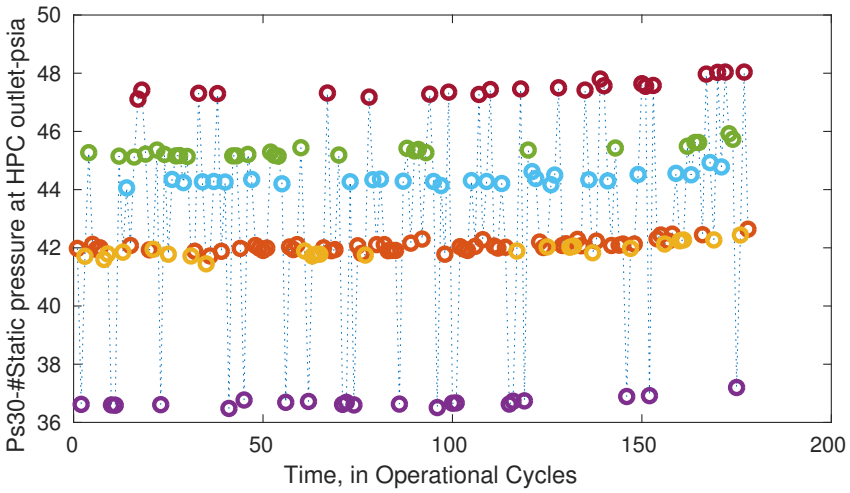

Figure 4.9: Clustered Regimes for Sensor 11 - Static pressure at HPC outlet

An illustrative example of the "raw sensor 11" after classification is given in Figure 4.9 (for further observations, see Appendix A.2 - Figure A.1). Compared to Figure 4.6, in which the plot's axis is limited to a certain data range, all six regimes are grouped effectively, and each regime is able to provide more meaningful information on system degradation.

As seen on the raw values from clustered time series, data are inconsistent and need to be adjusted from their original regime scales to a relatively common scale. The "multi-regime normalisation" method proposed in the methodology (section 3.4.1.2) adjust the data by bringing the all probability distributions of values into an alignment.

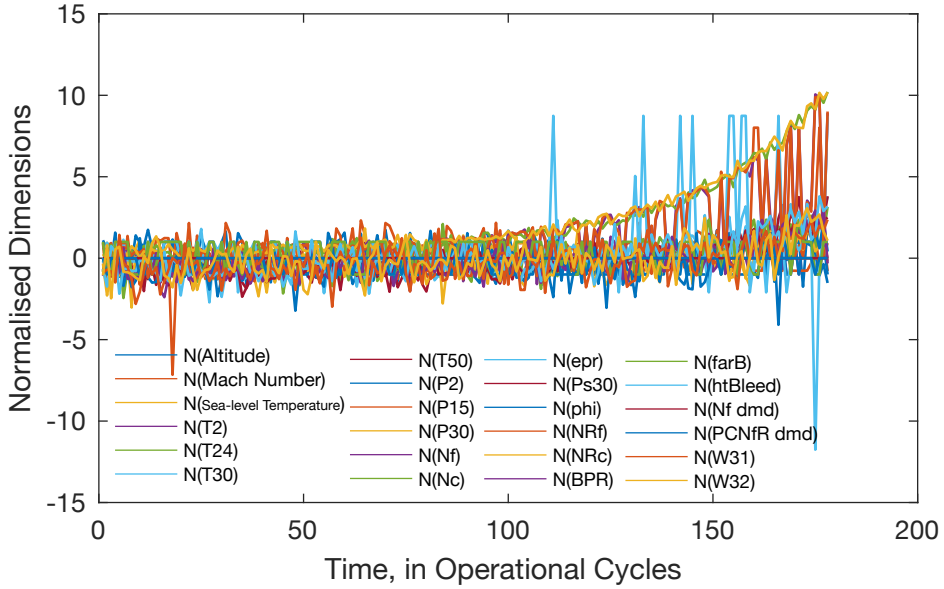

Figure 4.10: Normalised Sensors

In Figure 4.10, the standardised sensors after multiple regime normalisation are shown for a full operational period of a training trajectory. The regime differences are eliminated here, and the data is transformed into a form that allows for only one valid regime for all sensors. To evaluate how well these sensors show the useful degradation patterns, the prognostic parameter choosing measures of monotonicity, prognosability, and trendability reported in section 3.4.1.3 are applied to all training trajectories in this case study.

In accordance with the results of these metrics in Table 4.4, the ten sensors 2 (T24), 3 (T30), 4 (T50), 7 (P30), 11 (Ps30), 12 (phi), 15 (BPR), 17

(htBleed), 20(W31) and 21(W32) are considered meaningful for prognosis and will be employed in the following sections.

Table 4.4: Results of Prognostic Parameter Suitability Metrics

Sensor	Mon.	Prog.	Trend.	Sum		Mon.	Prog.	Trend.	Sum
1	0.25923	0.47787	5.00E-05	0.73715	**12**	**0.7908**	**0.7853**	**0.71302**	**2.28912**
2	**0.81147**	**0.82162**	**0.84246**	**2.47555**	13	0.47782	0.42358	8.00E-05	0.90148
3	**0.79834**	**0.81355**	**0.8238**	**2.43569**	14	0.58089	0.28385	0.00014	0.86488
4	**0.86291**	**0.86741**	**0.91356**	**2.64388**	**15**	**0.85703**	**0.8139**	**0.87213**	**2.54306**
5	0.24526	0.44186	0	0.68712	16	0.31476	0.47926	2.00E-05	0.79404
6	0.34362	0.53064	4.00E-05	0.8743	**17**	**0.82813**	**0.83043**	**0.84071**	**2.49927**
7	**0.78159**	**0.74244**	**0.70277**	**2.2268**	18	0	0	0	0
8	0.49842	0.42204	0.0001	0.92056	19	0.286	0.44186	1.00E-05	0.72787
9	0.55239	0.3148	0.00037	0.86756	**20**	**0.73925**	**0.76531**	**0.7017**	**2.20626**
10	0.25824	0.18214	1.00E-05	0.44039	**21**	**0.74419**	**0.76612**	**0.74673**	**2.25704**
11	**0.90393**	**0.89343**	**0.92564**	**2.723**					

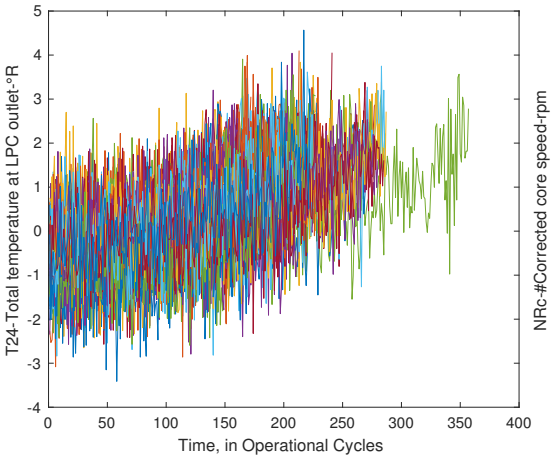

Figure 4.11: Useful sensors in all training trajectories

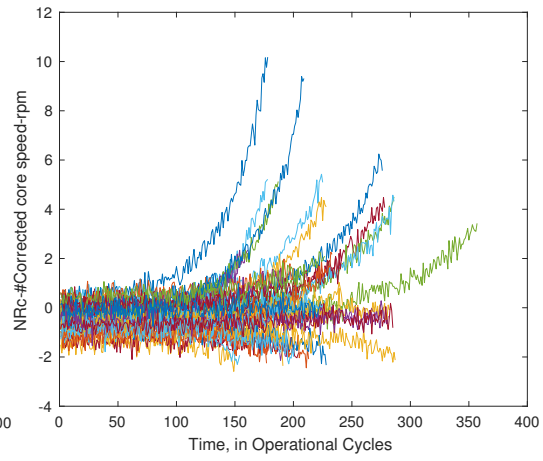

Figure 4.12: Non-Useful sensors in all training trajectories

Samples of useful and non-useful sensors are shown in Figure 4.11 and Figure 4.12. With respect to the fitness function as a sum of these measurements, the determined suitable sensors can also be visually observed and compared. In Appendix A.2, further sensor observations are provided for a

comparison between the prognostic parameter-choosing measures and the sensor behaviours.

Figure 4.13 demonstrates the (normalised) useful sensors to be applied in further stages. All these sensors show an exponential increase starting from an initial wear level and ending at a threshold point. Signals do not have smooth curves as they are contaminated with noise, but their quality can be improved by a suitable filtering method.

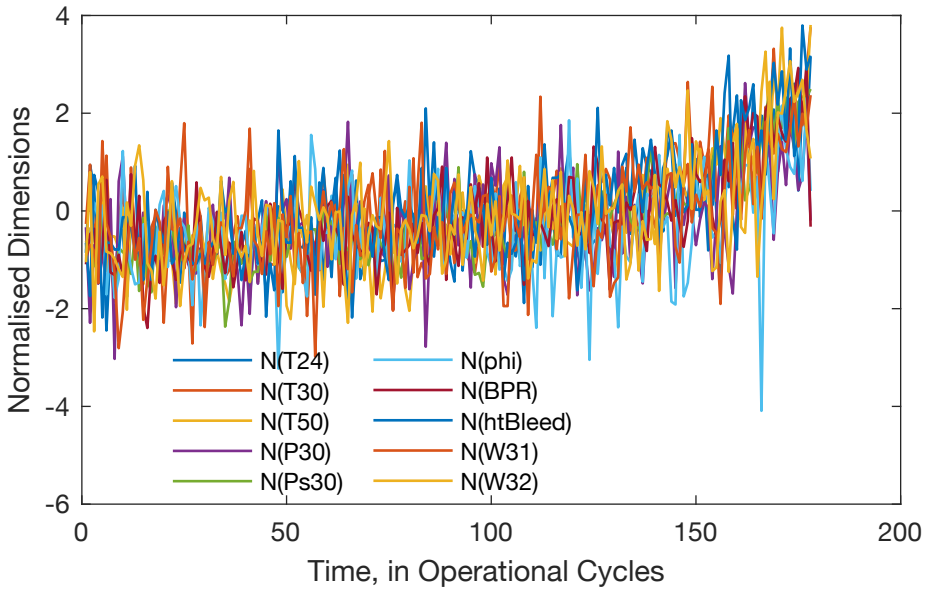

Figure 4.13: Useful Normalised Sensors

The normalised useful sensors are in a single homogeneous domain. However, as seen in Figure 4.13, these sensors are noisy and could perform poorly in terms of identifying the system's health degradation. It is necessary to emphasise the noise variations in the dataset and bring out strong patterns in the population. A smoothed and averaged version of an initial HI with respect to the normalised sensors is calculated. This filtering does not include a sliding window size to determine means. Instead, the mean over all population at the same time index (as described in the methodology chapter - section

3.4.1.3) is used to receive smoother curves.

The features extracted from the standardised sensor data are still contaminated with noise, and during the development of the methodology, it has been observed that there is always a risk that a neural network might learn from this noise in the training stage. In order to minimise this risk, a regression model is used to describe the relationship between the adjusted cycle index and health indicator so that the degradation curve is a smooth line and does not include errors. A two-term power series model, as given by the following equation is used to describe the final smooth HI parameter (MathWorks, 2002).

$$y = as^b + c + \varepsilon \tag{4.1}$$

where an approximation to a power-law distribution s^b has the two fitting terms of "a" and "c", which can represent uncertainty in the observed values. The main reason behind employing this fitting model is that the fitted HIs, y, for the run to failure trajectories demonstrate only increasing values and the initial operational stages behave in such a way that the fitting has stable wear levels with minimal increase. Assuming that the failure occurs at a certain stage of operation and the system degradation before this point is relatively steady (Heimes, 2008), the two-term power series model is able to define the hypothetical degradation effectively. Hence, to preserve the original degradation pattern, the two-term power series fitting model is used to identify a standard HI.

In Figure 4.14, a comparison of adjusted trajectory (mean at each time point) and fitted HI is illustrated. As can be seen from the blue curve, the degradation is stable up to a certain point, and the fit does not include any type of fluctuation.

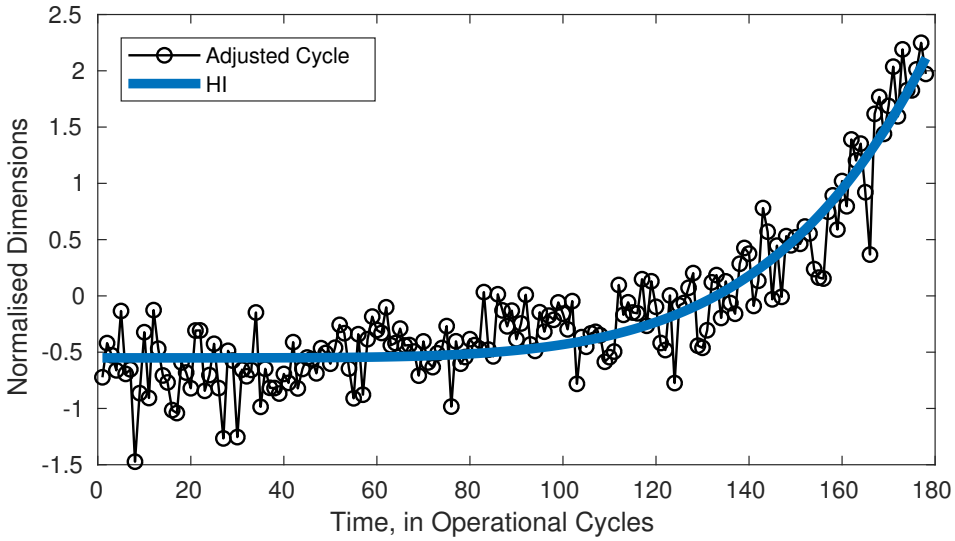

Figure 4.14: Adjusted Trajectory and HI

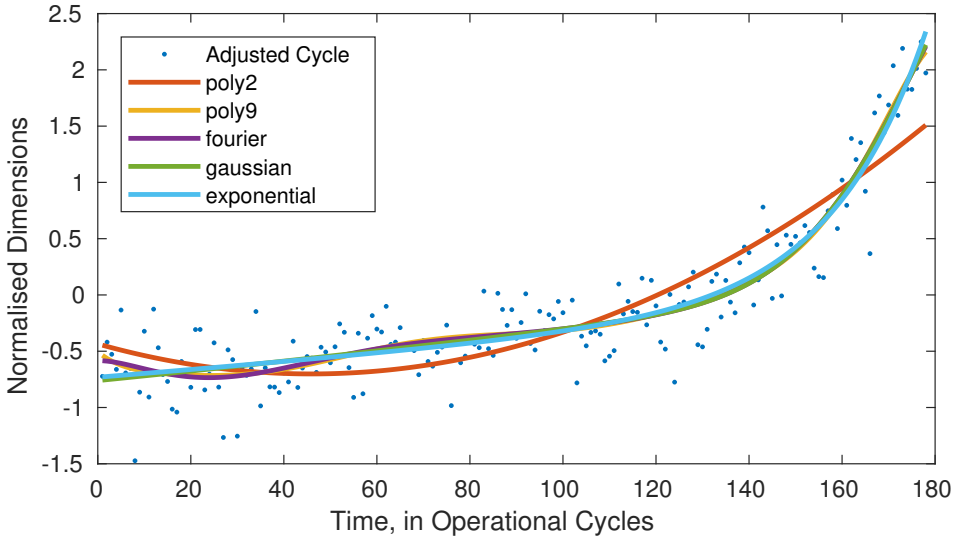

Figure 4.15: Alternative Fitting Functions

Alternative functions to power law model are shown in Figure 4.15. This example demonstrates how to fit the adjusted cycle curve to a set of fitting functions using Polynomial, Fourier transform, Gaussian and Exponential models (MathWorks, 2002). The fitted curves could help to identify the degradation level with time, but none of them is capable of defining the early stages of the HI as stable or steady.

4.3 Network Configuration

After the useful health features are extracted from raw data, the first step towards training the network is the configuration of input and output series. Inputs are situated as "$10 \times t$" cell of a matrix in which the dynamic data from 10 different variables of raw useful sensor data is represented. The target series is a "$1 \times t$" cell array of a matrix and only represents "t" time steps of an output variable which is the calculated HI from multi regime normalisation and feature extraction. The goal of this configuration is to provide the neural network mapping between a raw data set of numeric inputs and a set of calculated health indicators.

With an attempt to increase the deployment performance of the network configuration between trained inputs and targets, the multiple feed-forward networks are modelled in a double loop design over an increasing number of hidden layer of an outer loop and continuous weight initialisation of an inner loop. In other words, the hidden layers are not set to a default number. Instead, an outer loop of network training is arranged with an increasing number of hidden layers. Both loops terminate when a minimum desired error occurs with the overall generalisation. The optimum layer size is accepted according to the mean squared difference between the initial output series and the estimate. After these loops, the least erroneous network configuration is used to choose the best open loop design mode for validation.

In Figure 4.16, the configuration of network training stage is shown. The inputs are the raw sensor values for a single trajectory while the target is the calculated HI for the same trajectory. It is expected that function will be generated from the network structure that can map between these variables.

To prove that the trained Bayesian regularised neural network is capable of achieving high accuracy for an output estimation, the same raw data is added into the obtained network function to receive a neural network predic-

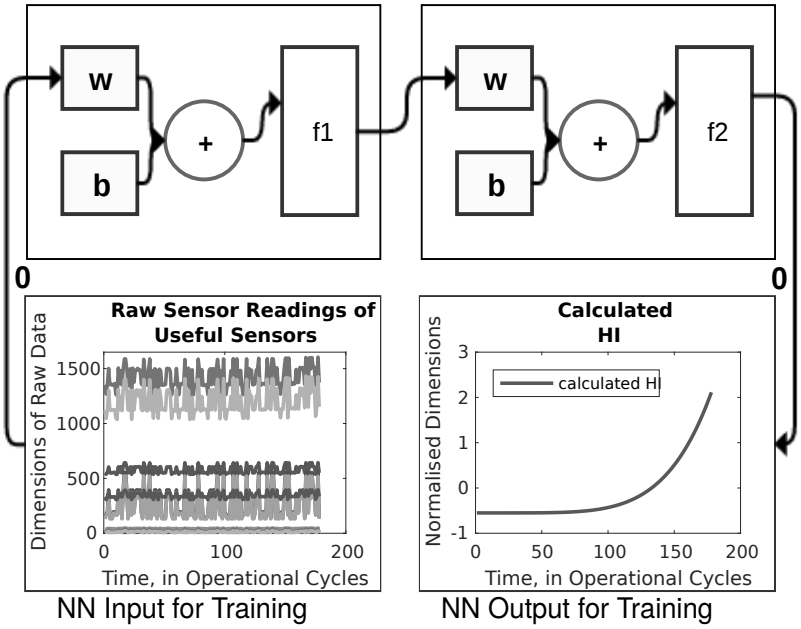

Figure 4.16: Neural Network Learning Stage

tion. There is no network training at this stage, and the same raw data used in training is applied to the network memory elements to receive an estimated output. Figure 4.17 represents the trained network functions rather than the network training and, accordingly their connection is marked with (1), rather than (0) as in Figure 4.16. According to the network estimation, one can confirm whether the proposed network model is accurate or if the network fitting meets the needs of prognosis.

As seen in Figure 4.17, the results show a pattern that is very similar to the fitted HI used in network training (see Figure 4.16) but as expected, it is contaminated with noise due to the generalisation of Bayesian regularisation algorithm. Both patterns follow exponential growth curves and their scales match each other. The initial wear level and threshold point of the estimation are coherent with the HI fit, and the output is normalised onto a common scale. The curve starts at $HI \approx< 0$ and reaches the failure point at $HI \approx> 2$. Comparing to the network training output in Figure 4.16, the result matches

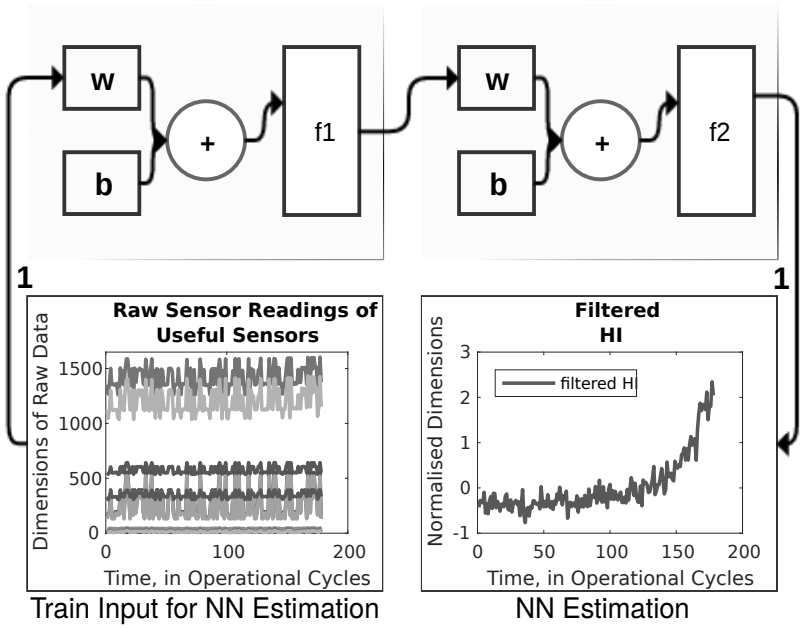

Figure 4.17: Neural Network validation with the same input

what might well be expected in a degradation model. Thus, it can be concluded that the network can effectively filter the raw data and standardise the outputs without requiring all the trajectories in the dataset. This achievement removes one of the major limitations of multi-regime normalisation of requiring all data at once. It will also lead to individual data filtering and normalisation for both training and test trajectories. Although the noise could be seen as the biggest variation between the initial training output and the network estimation, it could be regarded as a result of good generalisation for a novel input series to be submitted to the same trained network function.

In the case that the estimation is made for the same data used for training in Figure 4.17, it can be noticed that the response is satisfactory. However, as clarified in the literature chapter (section 2.3.3.3.3), overfitting and computational overheads might lead to poor network calculations in the training of multilayer neural networks. To validate that Bayesian regularisation algorithm

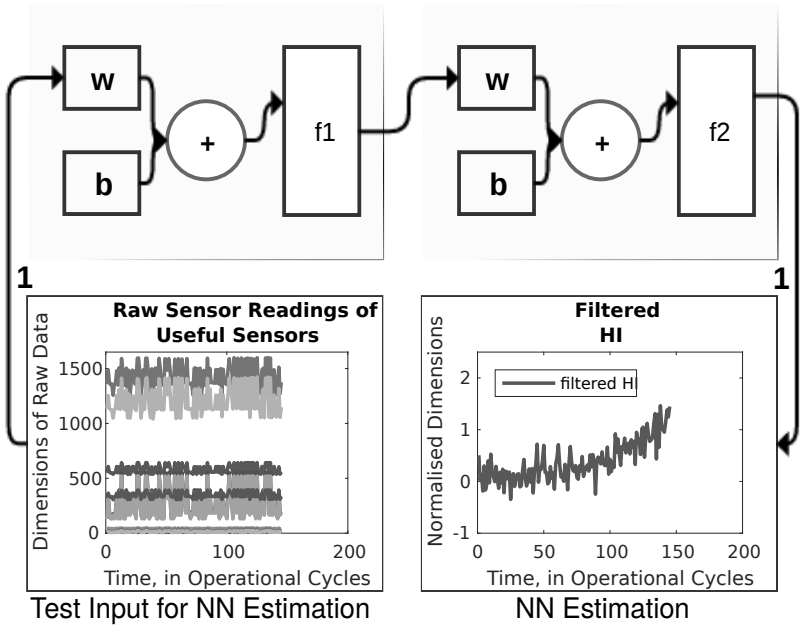

Figure 4.18: Neural Network estimation with a different input

in the feed forward neural network structure can memorise the samples for the training data and properly generalise the upcoming testing cases, the trained network function needs to be put to use on new inputs. The regularisation analysis of the network response could be performed with an alternative trajectory. In Figure 4.18, a raw test input with shorter instances and different degradation characteristics is used for estimation. Although the length and characteristic trajectory features (such as initial wear level and degradation pattern) are not the same, the model was able to accurately define the HI curve of the inserted data. In this instance, the curve starts at $HI \approx> 0$ and reaches a final point at $HI \approx> 1$. As the input data change, the HI output estimation is adapted to an appropriate time series, depending on the range of input sensor values. The neural network function trained with Bayesian regularisation, regardless of the data used in its training, could provide the intended results of HI estimations for different test trajectories, generalise the

upcoming novel cases with regard to their characteristics, and avoid overfitting and computational overheads.

The demonstrated Bayesian regularised network function is trained with a particular trajectory and could provide the intended results as seen in Figures 4.17 and 4.18. However, the regularisation and network training can also be further validated for the risk that if the function is regularised and trained with other raw input measurements and HI outputs, the estimations might result in different scale due to different weight and bias values. For this purpose, alternative neural network functions trained with different trajectories (inputs and outputs) are applied in the same raw training trajectory. They form a library for multiple calculations for each raw input.

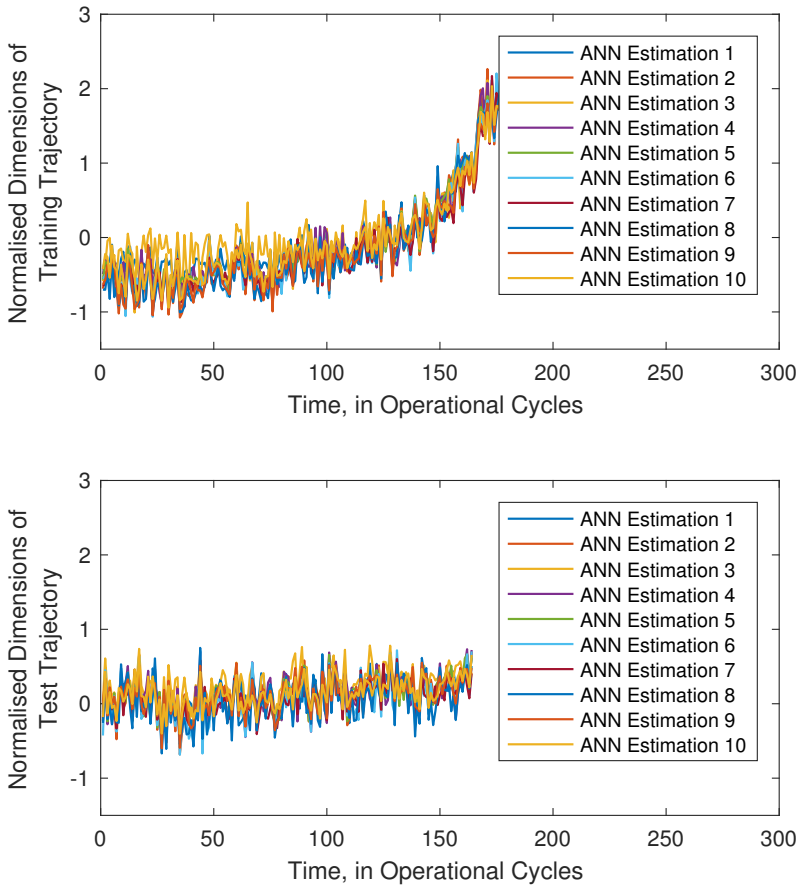

Figure 4.19: Neural Network Library Results

A graphical illustration of multiple network library results is given in Figure 4.19. The plot shows the network library estimations for a single training data with a full operational time index, while the lower plot shows the estimations for test data with a shorter time index. Each ANN estimation represented in the plots is a calculated output of the same input. The network library presents a set of network functions that are highly configurable and extendable. The results of distinct network instances are similar to each other, and the filtering library trained with Bayesian regularised feed-forward networks is found to assign the HI outputs adequately. It is worth noting that the neural network library used in the data filtering process must be trained by using different data from a similar source as in the C-MAPSS simulations. If the data sources are from different and unrelated domains, the neural network structures in the library are more likely to fail to provide high-accuracy results for data filtering.

As the library results in multiple ANN estimations for a single input (trajectory), the final HI estimation is estimated by the modified moving average model as described in the methodology (section 3.4.2.3).

In Figure 4.20, in accordance with the directional window of a temporary duration matrix, the moving average filtering is applied by moving horizontally over all network estimations. The mean is computed over the neighbouring elements at each time. Then, the window moves to the following time steps and repeats the computation. This sliding process continues until all time units are exhausted. If the matrix of network estimations is required to produce a smoother filtering output, the moving window size must be increased in order to provide an increased number of elements in the matrix, so that the noise factor caused by individual sensors is reduced. However, it has been observed that smoother curves can still be affected by fluctuations and may overfit the HIs.

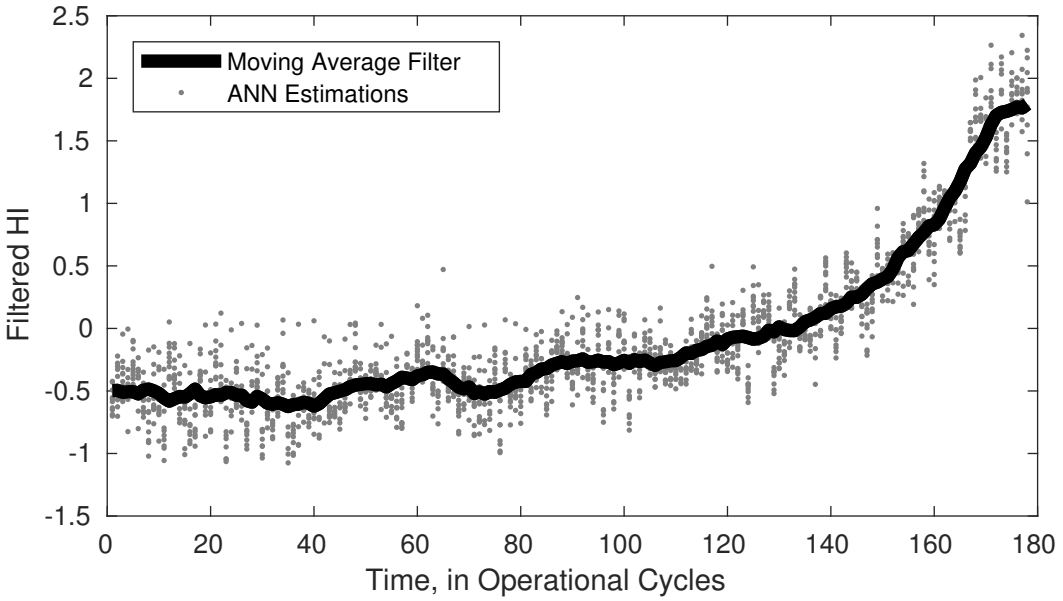

Figure 4.20: Moving Average Applied in Network Library Estimations

A further assessment of the Bayesian regularised network application and the moving average filtering method is achieved by interrupting the raw training input trajectories. Both full and interrupted data sets are inserted into the trained network functions separately, and then the moving average is applied to the network estimations from the library.

Figure 4.21 demonstrates a comparison of the first six raw trajectories in the dataset with their shorter forms. The concordance between the full and interrupted data demonstrate that the network library, even after moving average method has been applied, could map the raw data elements to the target vector in an organised sequence. Although full and short raw input series are inserted into the network function separately, the network estimations that can be achieved for the matching parts are nearly identical and only minor differences, which cannot cause major statistical deviations at the prediction stage, exist between the filtered outputs.

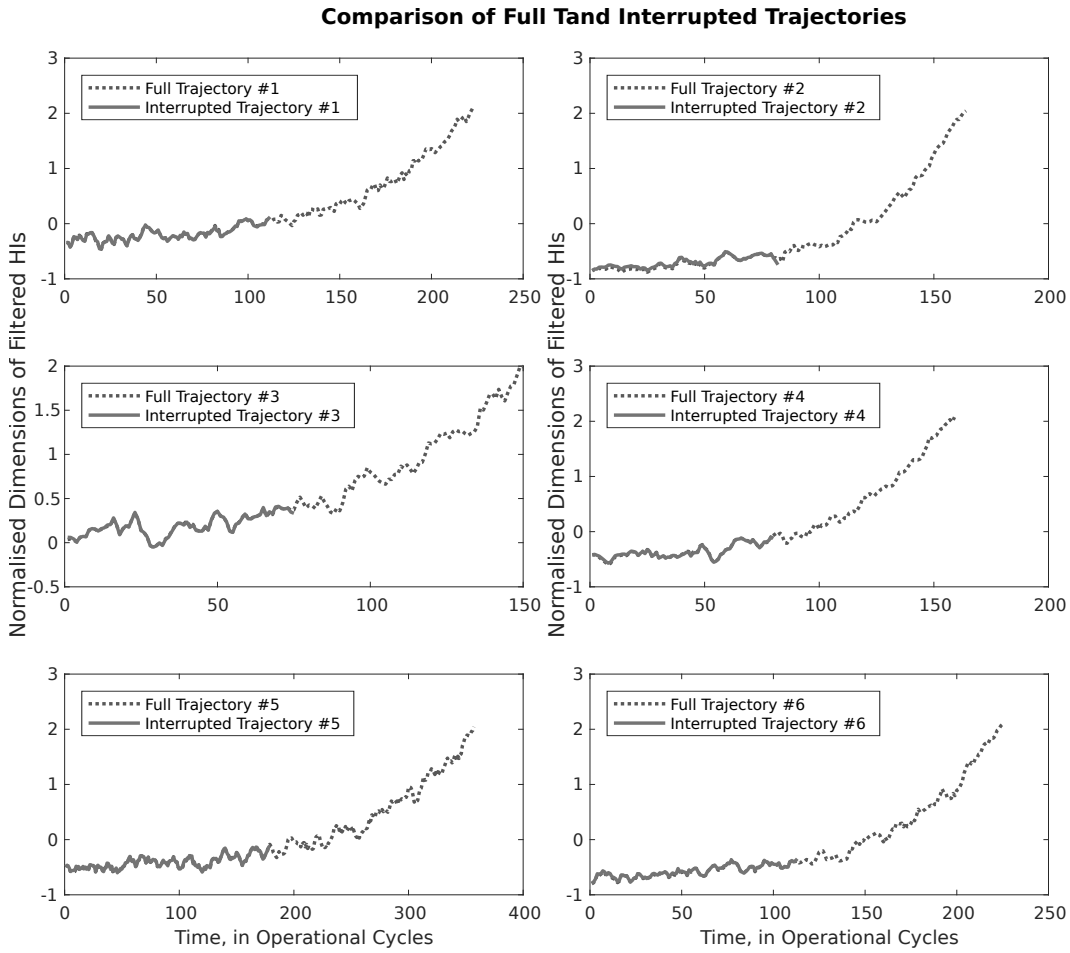

Figure 4.21: Comparison of Full and Interrupted Data

4.4 RUL Estimation

Once the neural network library provide a competent generalisation of the input-output relationship, the trained network functions can be used to filter all HIs for both training and test trajectories within the same domain. In the evaluation of many test instances, it is challenging to determine the likelihood of a failure point and the RUL. Test subsets have ended some time prior to the occurrence of any failure. When the operational length of data is sufficient and the degradation has matured enough to provide necessary information for RUL calculations, it can be observed in the experiments that the prediction

performance is satisfactory in terms of prognostic metrics. However, the short test instances, particularly the ones with less than 50 operational cycles, are at high risk of catastrophic failures.

Range of length (operational cycles)	≤ 50	$>50\& \leq 100$	$>100\& \leq 150$	$>150\& \leq 200$	$>200\& \leq 250$	≥ 250
Number of Test Trajectories	46	122	107	109	42	9

Table 4.5: Distribution of PHM08 Final Test Trajectories According to Operational Length

Table 4.5 classifies the PHM08 final test trajectories according to their length. Superficially, it could be argued that the prediction performance increases from the right to the left of the table. The main reason for this high risk at shorter sets is mostly related to the data interruption during a fluctuation caused by noise or other environmental settings. Although this fluctuation can be tolerated for mature instances, it is very likely to cause catastrophic failures in short instances.

With a lack of future data steps in the test data, the training subsets, which have demonstrated full operational life-times of engines and failed at a threshold point, is used to replace the missing future values of the test targets. In such a case, the multistep-ahead estimation method needs to determine which of the historical training data patterns is more likely to represent the test data.

When C-MAPSS data is considered, the functional mapping-based models (Heimes, 2008; Peel, 2008; Abbas, 2010; Rigamonti et al., 2016; Bektas and Jones, 2016) and the extrapolation-based models (Coble and Hines, 2011; Siegel, 2009; Liu et al., 2013b; Bektas and Jones, 2015) are the major RUL estimation alternatives to similarity-based prognostics (Ramasso and Saxena, 2014). Using a functional mapping process may result in difficulties in adapting to new situations, and it would thus be necessary to adapt model parameters

if the process evolves, as opposed to maintaining the use of static mapping. Methods within this category are mostly based on neural networks with different architectures. ANN multi-step predictions in prognostic applications can be quite challenging when only a few time series or little previous knowledge about the degradation process is available, and where the failure point is expected to occur in the longer term (Bektas and Jones, 2016). This is generally more evident in the time series showing exponential growth or decay. In extrapolation-based models, RUL is made by the estimations beyond the original observation range on the basis of its relationship with training data. The predictions of mature and long trajectories can be effectively made between known observations, but extrapolation is subject to a degree of uncertainty and the risk of significant deviations in cases where available observations are low. Considering the short test time series in Table 4.5 and the exponential characteristics of the engine data, the similar-based prediction model, which only uses the real degradation pattern, is applied for RUL estimations. Additionally, since the complete trajectories of a known dataset and the novel ones could be filtered by the network function, a collaborative estimation of RULs can be carried out by multiple participating instances.

In Figure 4.22, a pairwise distance between the training and test sets is initially measured at the first time step and the testing curve is moved to the end of base curve to find the minimum pairwise distance between these sets.

The measure of distance between each pair is stored at each time index and the best matching location is used as feedback to complete the future degradation progress of test trajectory. Figure 4.23 shows the location of minimum pairwise distance value as well as the non-useful information to be removed, and the remaining part to represent test trajectory's future behaviour.

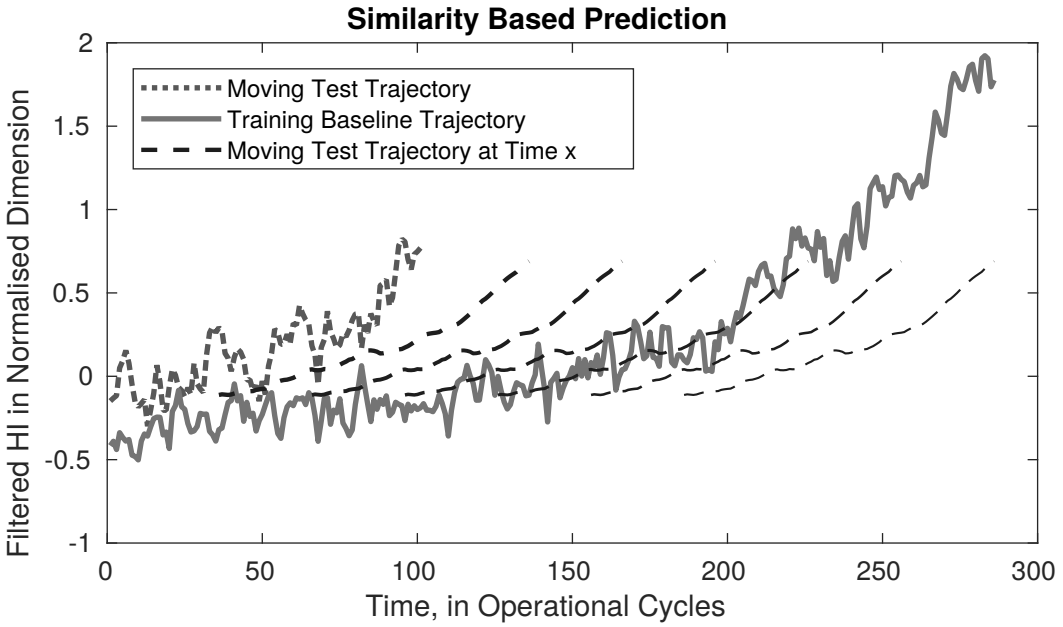

Figure 4.22: Pairwise Distance Calculation

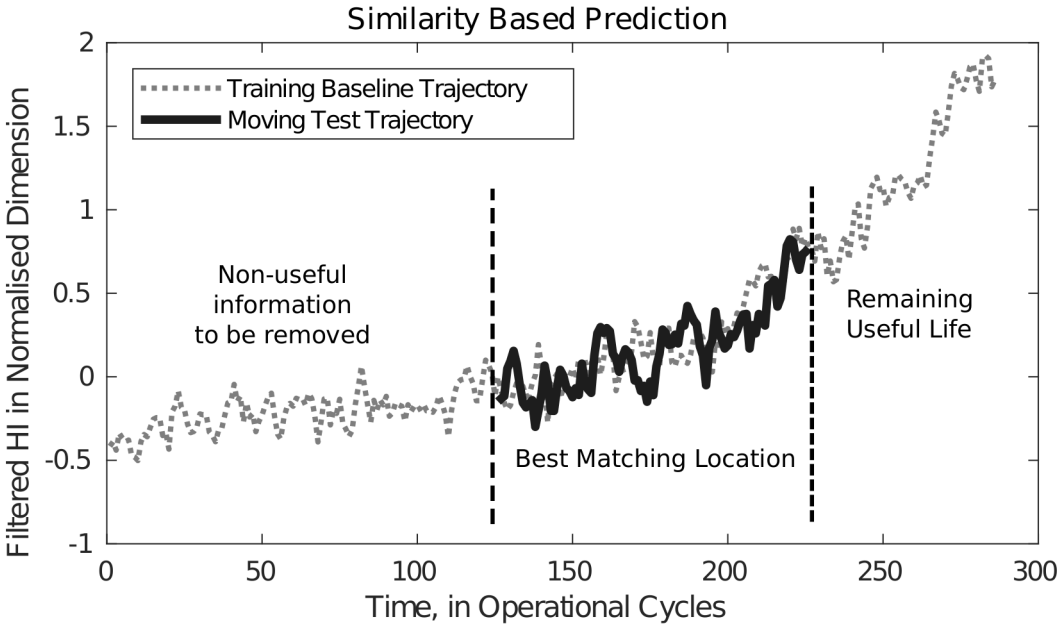

Figure 4.23: Minimum Pairwise Distance Position and Best Matching Location

A single similarity based RUL estimation with a single training baseline might be biased and produce an undesired result, no matter how similar it is to the test trajectory. However, this estimation model can be expanded to

other training trajectories and a data collaboration effort can be performed from multiple suppliers with a wide variety of degradation cases rather than using separate and self-contained sources. The calculations with more training data samples can increase prediction accuracy in multistep-ahead estimation. Therefore; a final RUL is estimated by the mean of corresponding RUL calculations of the minimum ten distances. In PHM08 challenge dataset, there are 218 training trajectories to be taken as baselines. Assuming that a test data is shorter than these baselines, the RUL fusion could be derived from the entire training dataset.

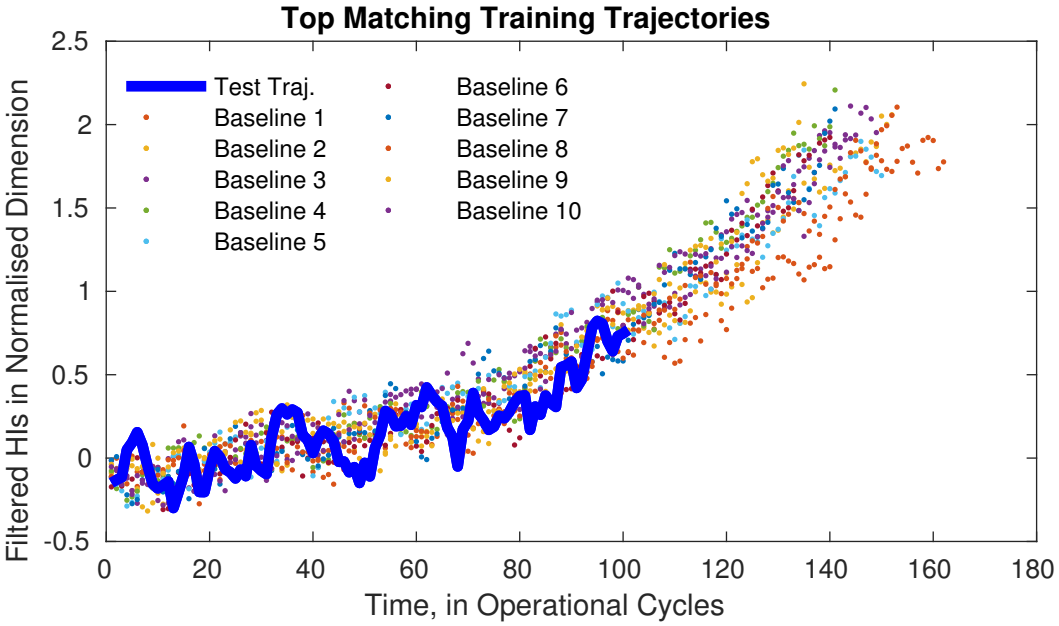

Figure 4.24: Top similarities and RUL calculation

Figure 4.24 shows the best matching trajectory parts for a single test instance. The degradation of test data here has grown so that the similarity-based estimation distribution is narrow enough to provide accurate calculations. All ten different baselines to be counted for RUL fusion follow relatively similar trends and their RULs are not different than each other. As one can assume, the error between the true RUL and the mean of these estimations will

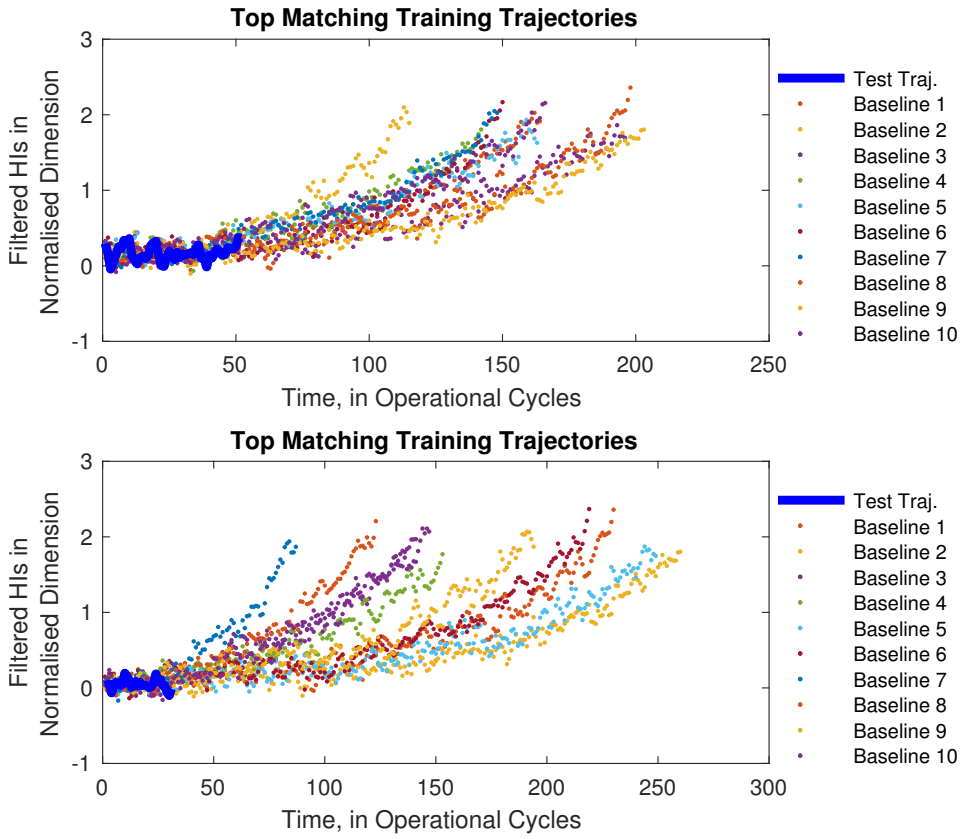

Figure 4.25: Test trajectories resulting in a wide range of RUL estimations

be small enough that the prediction is reliable in terms of prognostic metrics. However, as seen in Figure 4.25, the shorter test trajectories, or those that do not demonstrate a mature degradation pattern, are prone to providing wider distributions that can result in major deviations from the true RUL. The main reason for RUL fusion is to minimise the risk of such instances. Although some of the predictions made by the model cannot provide the intended results, the remaining will be of greater accuracy, and the mean of the population will avoid any potential catastrophic failure rate in the prognostic metrics.

4.5 Testing with Synthetic Data

Proposed framework could filter the C-MAPSS dataset trajectories into exponential health indicators and perform multistep ahead predictions. However,

degradation can occur in different forms and follow various patterns in different domains. To validate whether the framework could perform in such cases, the filtering method is tested with alternative synthetic datasets that follow different degradation patterns than the C-MAPSS dataset. Additionally, a synthetic target similar to the fitted health indicator in the Methodology (section 3.4.1.3) is used for an alternative network training.

Synthetic data in this section is primarily designed to correlate multiple sources of raw data derived from the C-MAPSS dataset with a predefined degradation curve. The neural network data filtering model is designed to identify any potential correlation that could provide alternative applications to multi-regime normalisation. Secondly, synthetic data is used to re-generate the subsets to evaluate the different wear conditions that can be found in various cases. This is particularly useful for understanding the performance of the network model's application to different degradation scenarios. It also allows one to account for unexpected conditions and to have a basic solution in any case where multi-regime normalisation might prove to be unsatisfactory.

Once a mathematical model for the synthetic data has been established, this is available for production of an adaptive target data for both training and validation. Since the exact behaviour of degradation change is known, the network with synthetic data could be evaluated with regards to the model, and the differences caused by the noise between the network estimation and the inserted target could be precisely identified.

With reference to the generalised time varying health index function (equation 2.31) introduced by Saxena et al. (2008b), Ramasso (2014a) set the following equation for synthetic HI for a corresponding target assignment:

$$\mathrm{sHI}_t = \exp\left(\frac{\log(0.05) \times (l-t)}{0.95 \times (l-1)}\right) \tag{4.2}$$

where t is the time unit, and l is the length of time series representing

the full sets of operations. This function forces the health indicator index to increase from pre-defined levels (such as 0, healthy to 1, faulty). Since there is no component-wise z-score normalisation applied for this equation as defined in chapter 3, the range of pre-defined levels cannot be known and therefore the synthetic HI (used for benchmarking) has a different range for the ones in previous sections of 4.2, and 4.4.

The application of a target generated from a synthetic HI function and the corresponding raw data trajectory from the PHM08 training set is configured and trained in a feed-forward neural network filtering model as explained in section 3.4.2. Since the HI output is generated by the network function, the clustering, regime identification, multi-regime normalisation, re-assembling and HI fitting steps described in existing failure mode (section 3.4.1) are replaced with the synthetic HI. After the network training is complete and all network memory parameters have been determined, the network can be deployed to estimate the target documents for filtering.

Figure 4.26 shows the comparison between the network estimation and the synthetic target. Although it is not expected that the synthetic target could precisely demonstrate the engine degradation behaviour, it could effectively help the neural network to filter the data. For initial stages of the operation, the network estimation could identify a relatively linear pattern despite the fact that the function has a constant exponential increase through the operational time units.

Developing a network library with synthetic targets, however, has apparent obstacles regarding initial wear level identification and final threshold point identification. Since all targets have to be assigned with a predefined range ([0,1] rather than the component-wise normalisation range in existing failure mode), each training would be designed by an output with the same range and the estimations based on these trainings would be standardised in

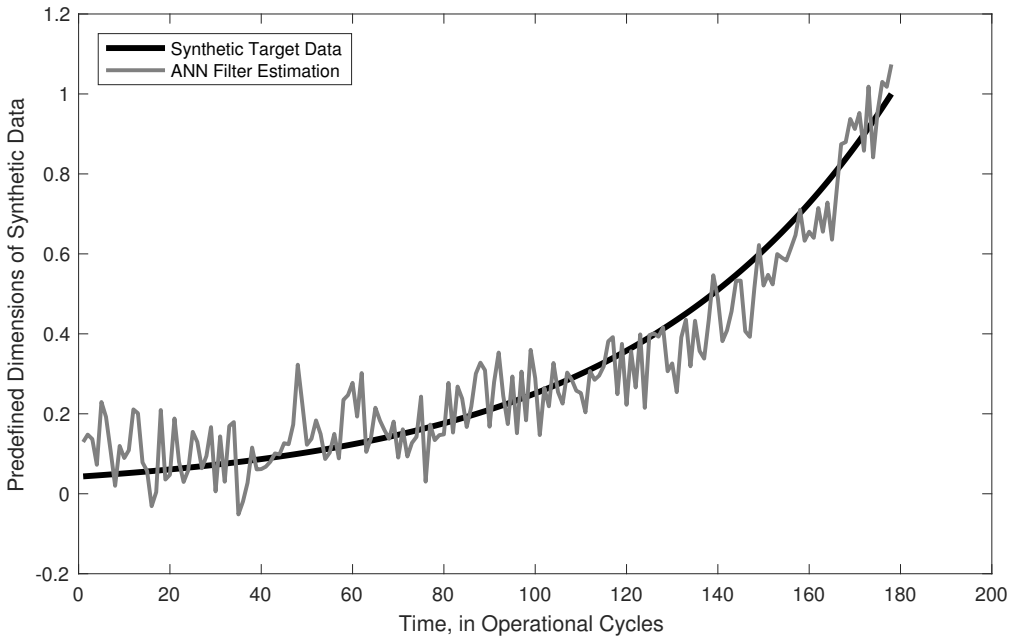

Figure 4.26: HI Estimation from a Network Trained using a Synthetic Target

an irregular manner due to the mismatching memory elements in the different networks. To overcome this issue, it can be argued that either an addition initial wear level identification model with feature scaling for the synthetic target can be applied, or the estimations can be based on a single network rather than a library with multiple networks. These applications can be particularly useful when the dataset is highly complex and thus not applicable for regime identification. In such cases, a population-based normalisation process may not satisfy the requirements of a meaningful HI identification, or the operational settings to identify the regimes may be intertwined. Since the synthetic HI does not require any data pre-processing steps, the neural network model with synthetic data training can be applied to fit the HIs for the purposes prognosis.

As demonstrated in Figure 4.26, obtaining the network filtering using synthetic data has provided a robust validation for the network adaptability. The target could be accurately estimated, and the network can identify

degradation behaviour differences between the raw data characteristics and the output of the function. On the other hand, the synthetic targets will lack the ability to form a network library in most cases where the regime clustering is available and component-wise z-score normalisation is easy to apply.

4.5.1 Network Training with Synthetic Data

The second application of synthetic data is used to re-generate the raw data values with alternative degradation patterns so the developed network model can be tested under different scenarios. These synthetic data models are generated to simulate particular conditions that may not be found in the PHM08 data sets. The produced dataset represents the authentic data and allows alternative baselines to be set. This enables the network model to learn from distinct behaviour profiles for filtering. The produced synthetic data is used to train the faulty system itself; thus, this application makes the required adaptations to the system required for distinct degradation environments.

In complex systems degradation, it has been observed that some patterns of degradation are linear (Li and Nilkitsaranont, 2009). To form such a baseline for data re-generation, equation 4.2 is first modified to model a linear decrease as:

$$sHI_t = 1 - \frac{l - t}{l - 1} \tag{4.3}$$

and secondly, a damage model similar to the exponential decreasing degradation pattern proposed by An et al. (2013) is formed as:

$$\text{sHI}_t = \exp\left(-\frac{\log(0.05) \times (l - t)}{0.95 \times t}\right) / \max(\text{sHI}) \tag{4.4}$$

In order to apply these functions to the dataset, the sensors are clustered for each regime. Then, a model is used to plot an arbitrary curve that identifies

the performance of sensor within the same regime. Due to the environmental settings and snapshot representation of the system, the exponential growth of the wear over the time series is very noisy. A filtering process is needed to simulate the scattered information that would be otherwise ineffectual or confusing to the performance index. To this end, a polynomial fitting model is used to identify the nonlinear relationship between the noisy raw values and the corresponding dependent variables. Comparing to other fitting methods such as Fourier transform, Gaussian and exponential models (MathWorks, 2002), the Polynomial method could more accurately identify the fluctuations in the time series, and therefore can provide less biased and more precise conclusions regarding noise factors.

The general polynomial regression model, when generalising from the standardised vectors to a k^{th} degree polynomial, is represented by the following equations (Anderson, 2011):

$$y_{f_i} = a_0 + a_1 x_{z_i} + a_2 x_{z_i}^2 + \ + a_k x_{z^i}^2 \quad \text{(i=1,2,...,k)} \tag{4.5}$$

where y is dependent variable, x_z is independent variable and a is response vector. This can be expressed in matrix form as:

$$\begin{bmatrix} y_{f_1} \\ y_{f_2} \\ y_{f_3} \\ \vdots \\ y_{f_n} \end{bmatrix} = \begin{bmatrix} 1 & x_{z_1} & x_{z_1}^2 & \cdots & x_{z_1}^k \\ 1 & x_{z_2} & x_{z_2}^2 & \cdots & x_{z_2}^k \\ 1 & x_{z_3} & x_{z_3}^2 & \cdots & x_{z_3}^k \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{z_n} & x_{z_n}^2 & \cdots & x_{z_n}^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix} \tag{4.6}$$

The equation in pure matrix notation is also written as:

$$\vec{y_f} = \mathbf{X}\vec{a} \tag{4.7}$$

A smoothed, filtered and cleaned degradation curve is achieved after the polynomial regression is received. The clustered regime parameters are subtracted from this curve to preserve the noise factors which have a significant effect on the realistic simulation. If a higher or lower degree polynomial regression is used, the scale of the noise might be misrepresented. Thus, a fourth-degree polynomial is applied to represent an arbitrary curve of the performance index.

The synthetic outputs of the functions are scaled according to the maximum and minimum of the arbitrary curve, and the subtracted noise factors are added to this scaled output. This is applied to all sensor clusters in each regime to change the main degradation pattern in all sensors. Figure 4.27 shows an example of the regenerated data. Although it is hard to observe the change in the raw data form, the regime clusters provide a better illustration of the regenerated data characteristics.

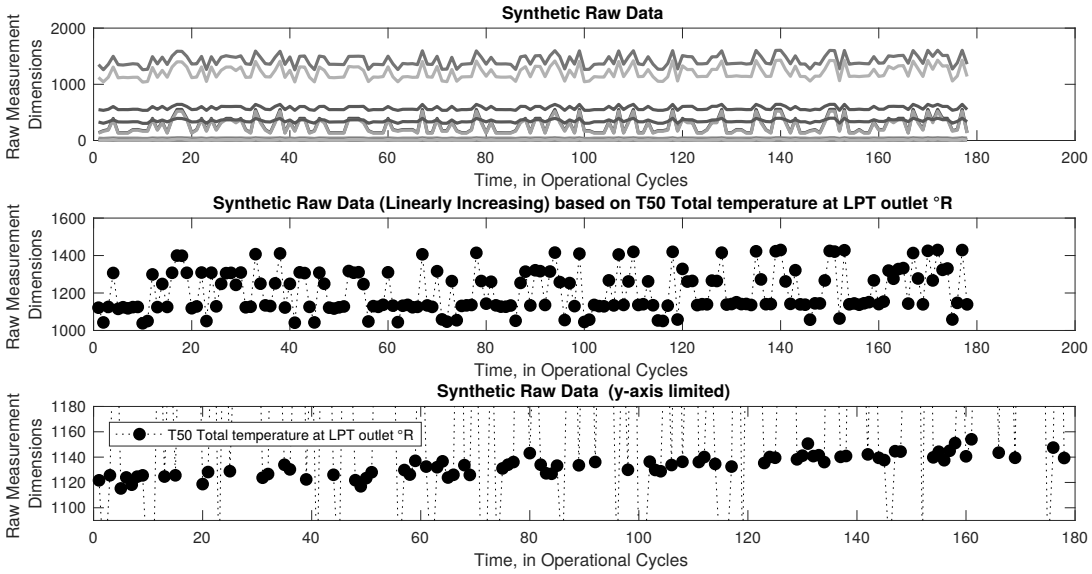

Figure 4.27: Synthetic Raw Data with a Linear Decrease

The exponential wear growth can be assumed to be equivalent to the parameters of aircraft engine modules such as efficiency and flow levels, and

can be generally linked with the performance of the main engine components. However, the failure mode characteristics for a different system might be distinct from this form of degradation. To assess the performance of the method developed in Section 3.4.2.1, the neural network filtering model is applied to regenerated synthetic raw datasets to provide a final validation. The initial function outputs are accepted as the target values for the network output. The generated training data sets that correlate with the wear growth pattern of the target are accepted as input series. The relationship between the synthetic outputs and network estimations are shown in Figure 4.28. The model convincingly filtered the datasets formed with various performance degradation patterns. Although the noise factors were added back into the data, the network estimation gave intended results for the outputs. In both plots, estimations match with their initial synthetic degradation patterns and the network filtering were capable of filtering data in alternative HI environments.

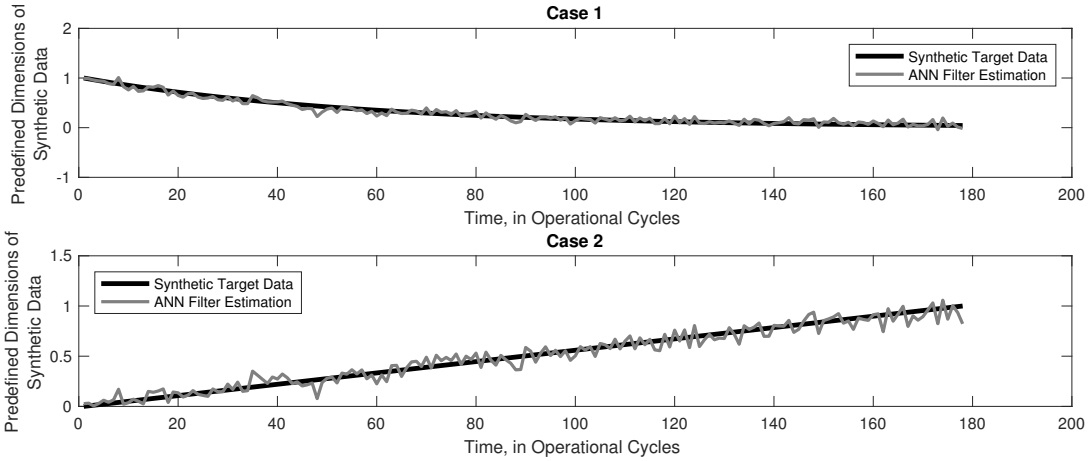

Figure 4.28: Network Estimations for Synthetic Data

In Appendix B, various data generation code fragments required for a realistic synthetic data generation process are provided. All can be viewed as prototypes for synthetic pattern formation and data modification.

4.6 Performance Pre-Evaluation Using Secondary Datasets

Before the results are sent for performance evaluation, in order to pre-validate the dataset with true RUL values, the algorithms are tested on secondary datasets derived from the noisy degradation patterns of original trajectories. Secondary data in this research refers to data that was collected from the training trajectories. Its use provides a robust validation of the model developed for the PHM08 data challenge that would otherwise be ambiguous due to the high risk of one-time submission. In the case of performance evaluation in particular, secondary data could provide increased knowledge that could feasibly allow for any re-assessment and re-development of algorithm steps. Furthermore, the developed model could be tested in terms of whether its application is compatible with different datasets representing different types of degradation. For example, the leading score (calculated by the score function metric in 2.4) for the PHM08 "test" data to date stands at 436.84, but the high performance in the "test" score does not guarantee the same performance level in the "final test" set. The model with the best test score, for instance, only ranked 22^{nd} on the final test leader board (Saxena and Goebel, 2008a). Since the full operational periods for the training data are known and its true life time can be calculated, secondary data can be applied to generate many realistic test scenarios for prognosis, and it can also provide a robust performance evaluation method for the validation of the accomplishments of designed algorithms prior to testing on the "final test" submission.

The total test trajectory unit number and each unit's length in the secondary data are identical to those of the original data set. For each secondary test unit produced, randomly selected original training data have been assigned as a base case. A randomly designated location from this base data

is then taken off and stored in the secondary dataset file. This means that there is no modification in the data and all values are originally sourced from the C-MAPSS simulation.

There are twenty different secondary datasets generated and used in the pre-validation of the final test subset for both filtering and prediction. Since they are produced from real training data, their failure point and, therefore their exact true RUL, is known. They contain real multidimensional and noisy sensor values within multiple regimes, and moreover, they are very practical and suitable for the developed model.

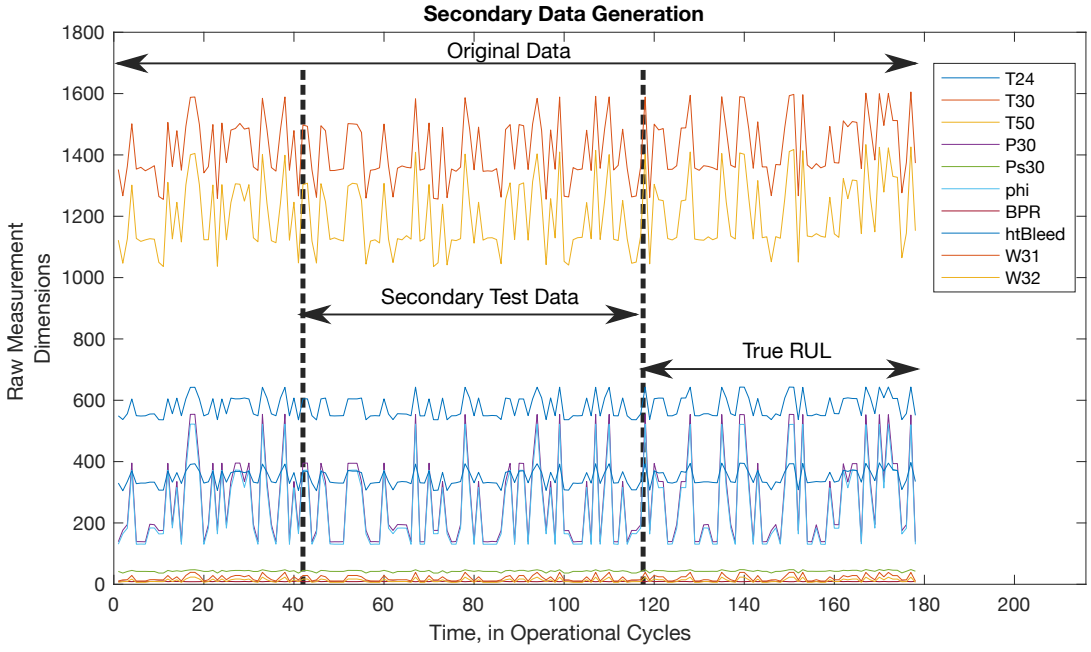

Figure 4.29: The principal start and end margins of secondary data

Figure 4.29 shows a sample of the secondary data location: the start and end margins for a test trajectory. This implementation aims to illustrate a further case study for the proposed model by using real degradation data from the C-MAPSS simulation. Based on the true RUL and full operational behaviours in the plot, the prognostic-related behaviour could be observed by

using the complete degradation pattern for each test unit. The continuation of selected location is available from the base training data and has already been sourced from the same supplier, making it convenient to carry out further research.

The location selection process for secondary datasets should be reliable and performance-efficient: the data was originally collected by someone other than the researcher, and the datasets may cover both larger and shorter samples of the population. In terms of prognostic metrics, a clear benefit of selecting training data with a random operational index is that it limits the effect of the participant and the validation work needed before submission can be carried out in multiple complex cases. The randomly selected location illustrates the possible high-risk factors, and the potential range of performance scores for the final test subset can be estimated by using pre-evaluated datasets. However, it was observed in the original data that the short "final test" trajectories were always found in the early stages of the degradation. To be sure that a short secondary data instance could provide a baseline for primary research design, a maximum ending point for random selection was set to determine the location accurately. Thus, the short test trajectories might have a pre-established level of reliability.

MATLAB code with a step-by-step explanation of secondary data generation procedures is given in Appendix C.

4.7 Critical Discussion of Model to Other Data Sources

A prognostic data source is expected to include a minimum sample size to allow an effective modelling on condition monitoring information of systems. The methods use the degradation patterns of sufficient samples representing

system failure progression. This is the major challenge in prognostics as it is often not possible to have multiple samples at once and industrial systems are not allowed to operate until failure due to the catastrophic consequences in critical systems (Eker et al., 2012). However, a variety of condition monitoring datasets have been published such as bearing data set of Lee et al. (2007) , battery data set of Saha and Goebel (2007) , turbofan degradation simulation data set (Saxena and Goebel, 2008b), thermal overstress accelerated ageing data set (Insulated Gate Bipolar Transistors - IGBT) of Celaya et al. (2009), bearings' accelerated life tests provided by FEMTO-ST Institute (Nectoux et al., 2012) and Virkler crack growth data set (Virkler et al., 1979).

It would be challenging to develop a data-driven model in the cases where run-to-failure sets of samples are limited such as in the bearing dataset with three sets of run-to-failure simulations Lee et al. (2007), battery data set with four set of samples under the same operating and environmental conditions Saha and Goebel (2007), IGBT with five run-to-failure samples under same conditions (Celaya et al., 2009), FEMTO bearing data set with six run-to-failure datasets (Nectoux et al., 2012). Limited training sets are not enough to apply data-driven prognostics in an effective way and one should consider the physics-based modelling for such cases (Eker et al., 2012). However, as mentioned in the literature review (section 2.3.1), the physics-based prognostics are mainly designed for a specific domain and are not generally applicable to a different applications without a significant amount of effort. Besides, they are computationally expensive, component specific and difficult to apply in complex systems operating under multiple operational conditions and producing multidimensional and multivariate signals from its sensors.

To generalise the proposed application, the datasets with sufficient run-to-failure sets of training samples, such as Virkler data set with 68 (Virkler et al., 1979) specimens and turbofan data with at least 100 degradation simu-

lations for each set (Saxena and Goebel, 2008b) are major alternatives. These information sources are eligible for data-driven prognostics, as there are sufficient samples to train the models (Eker et al., 2012). A minor issue in these is the lack of third-party validation of results as like the case of PHM08 (Saxena and Goebel, 2008a).

In this section, the discussion of the proposed framework to other data sources can be further expanded by the assumptions on the feasibility of potential condition monitoring data which is summarised as below:

- The monitored system is a group of interrelated and interacting constituents forming a complex whole.

- Condition monitoring is based on various sensors of components and provides information on the system health status.

- Sufficient amount of training data to is available to allow an effective prognostic modelling.

- Data is monitored at discrete time such as operational cycles. The health status of the system degrades over measurement cycles.

- The system's health degradation can be estimated from the existing run-to-failure training instances.

Considering a data source has these characteristics, the proposed methodology has a potential to estimate an unknown RULs effectively and to predict the health degradation in multi-step long-term cycles. The deficiencies exhibited by multi regime conditions can be removed through the hybrid model of regime normalisation and neural network data filtering.

As this adaptive filter model is applied in trajectory similarity-based multi step ahead estimations, the framework can provide a collaborative RUL estimation model with high performance results.

4.8 Summary

This chapter presents how the prognostic framework proposed in methodology chapter can be applied in a complex system domain with multidimensional condition monitoring data under multiple regimes.

A case study of the presented methodology is defined in gas turbine engine domains. Considering the main reasons why gas turbines are appropriate applications for the framework and how they exemplify a complex system, the chapter introduces the characteristics of gas turbine systems and engine health monitoring. Due to the prognostic requirements for common datasets and a mutual comparison to validate the methods introduced by different researchers, "PHM08 Challenge Data Set" is used in the case study. The proposed methodology proved its dimensionality reduction and RUL estimation capabilities through the use of simulation data. Furthermore, the filtering framework is first benchmarked with a synthetic HI adaptation model, and then it is tested with re-generated raw data with alternative degradation patterns so that the developed network model can be applied under different scenarios. A final validation is suggested through secondary datasets which refer to test data that was collected from the training trajectories. Use of secondary data could provide increased knowledge that could feasibly allow for the application.

In conclusion, the proposed prognostic concept in methodology chapter proves to be promising to be applied in a complex system operating under multiple regimes. The framework was able to identify the failure degradation in multiple domains at any time index of the operational cycles. The challenge was to normalise these failure information from multiple regime domains into a single domain as well as preserving the characteristics of trajectories with regard to each other and allowing the system to operate in novel cases without a system restart. The framework could overcome these challenges and provide

filtered HIs for a collaborative RUL estimation.

Chapter 5

Results & Discussions

This chapter presents the results obtained from the prognostic framework and case studies mentioned in the previous chapter. RUL estimation results are mainly analysed for the final test subset of the PHM08 data challenge (Saxena and Goebel, 2008a) and twenty different secondary datasets derived from original training data. Although the validation can be expanded to other datasets of the C-MAPSS simulation and potentially to other multidimensional condition monitoring data, this research compares the results for the "final test data". This is mainly because the true RUL is not provided for the PHM08 data challenge and the metric evaluation can only be completed by the NASA PCoE (rather than the author). Therefore, a model validation with this data is much less likely to be biased toward a single individual dataset case and can confirm whether the methodology can be expanded to other complex systems operating under different regimes.

The following sections present pattern of performance results and analyses them for their relevance to the research objectives. The results reveal that the proposed data filtering processes for collaborative RUL estimation is able to increase the performance of RUL estimations. As compared to the published works, it achieved the overall leading score in the published literature.

5.1 Prognostic Metrics for Analysis

Most prognostics concepts lack a general agreement as to a universal set of metrics that can be employed to assess the performance of prognostic systems (Vachtsevanos et al.). This lack of agreement is mostly due to the various user requirements for different applications such as time scales, available system information, domain dynamics etc. (Saxena et al., 2009a). However, as given in chapter 2, the research community has introduced a variety of metrics to evaluate critical aspects of RUL predictions before they are applied in real-life domains. These metrics form a standardised method of communication by which the users can be demonstrated and compares findings (Goebel et al., 2011a). Considering that such communication allows the suitable expression of scientific information, the metrics given in section 2.4 are adapted for performance analysis of this research (see table 5.1).

A particular importance between these metrics is given to scoring function due to its exponential characteristic on penalising high error rates more than lower ones. For a degradation scenario of complex systems such as gas turbine engines, it is more critical to assess the performance of RUL estimations. Considering the increasing risk with error rate, the scoring function is applied to evaluate the RUL estimations. In the case studies, each data point is a snapshot so that the run-to-failure time series are mostly less than 300 time steps. As mentioned in the literature chapter (section 2.4), the user-defined parameters of scoring function should be assigned in accordance with the length of condition monitoring time series so that the results are not biased against the error growth.

One further aspect is that the late predictions in gas turbine engines are more risky than the early predictions due to the potential catastrophic rates (Saxena et al., 2008b). Therefore, the scoring algorithm for performance evaluation is specified to be asymmetric around the true RUL such that late

Table 5.1: Prognostic Metrics (Saxena et al., 2008a; Goebel et al., 2011a; Hyndman and Koehler, 2006)

Metric	**Formula**	
Mean Absolute Error	$\text{MAE} = \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} \lvert T_i - P_i \rvert$	T=true (known) RUL P=predicted RUL n_{te}=number of test trajectories
Mean Absolute Percentage Error	$\text{MAPE} = \frac{100}{n_{te}} \sum_{i=1}^{n_{te}} \left\lvert \frac{T_i - P_i}{T_i} \right\rvert$	
Mean Square Error	$\text{MSE} = \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} (T_i - P_i)^2$	
Scoring Function	$s = \begin{cases} \sum_{i=1}^{n_{te}} e^{-\left(\frac{T_i - P_i}{a_1}\right)} & \text{for} \quad T_i - P_i < 0 \\ \sum_{i=1}^{n_{te}} e^{\left(\frac{T_i - P_i}{a_2}\right)} & \text{for} \quad T_i - P_i \geq 0 \end{cases}$	a_1,a_2= user-defined acceptable early and late parameters
Sample Standard Deviation	$S = \sqrt{\frac{1}{n_{te} - 1} \sum_{i=1}^{n_{te}} ((T_i - P_i) - \mu_e)^2}$	μ_e=sample mean of error (T-P)
False Positive Rate	$FP(i) = \begin{cases} 1 & (T_i - P_i) \geq t_{FP} \\ 0 & \text{otherwise} \end{cases}$	t_{FP}, t_{FN}=0
False Negative Rate	$FN(i) = \begin{cases} 1 & -(T_i - P_i) > t_{FN} \\ 0 & \text{otherwise} \end{cases}$	
Mean Absolute Deviation from Sample Median	$MAD_i = \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} \lvert (T_i - P_i) - M \rvert$	$M = median(e)$ median is $\frac{n+1}{2} th$ order statistic
Median Absolute Deviation from Sample Median	$\text{MdAD} = median\left(\lvert (T_i - P_i) - M \rvert\right)$	

estimations are more heavily than early predictions. Considering these points as well as the lack of standardisation and common comparing analyses, the "user-defined parameters" are specified according to the works of Saxena et al. (2008a) as to be $a_1 = 10$ and $a_2 = 13$. This allows the benchmarking of the presented methodology with rivals who commonly use the same parameters to validate and compare their works.

5.2 Performance Evaluation

The purpose of performance evaluation is to interpret and describe the significance of developed model in light of what was already known about the

prognostic problem, and to explain any novel understanding or insights gained towards the problem after the findings are taken into consideration. To that end, the results from secondary datasets are first used for a pre-evaluation of the model. Since true RULs are known from the original run-to-failure data, error based prognostic metrics can be applied and the performance of the framework can be assessed by using known life-time.

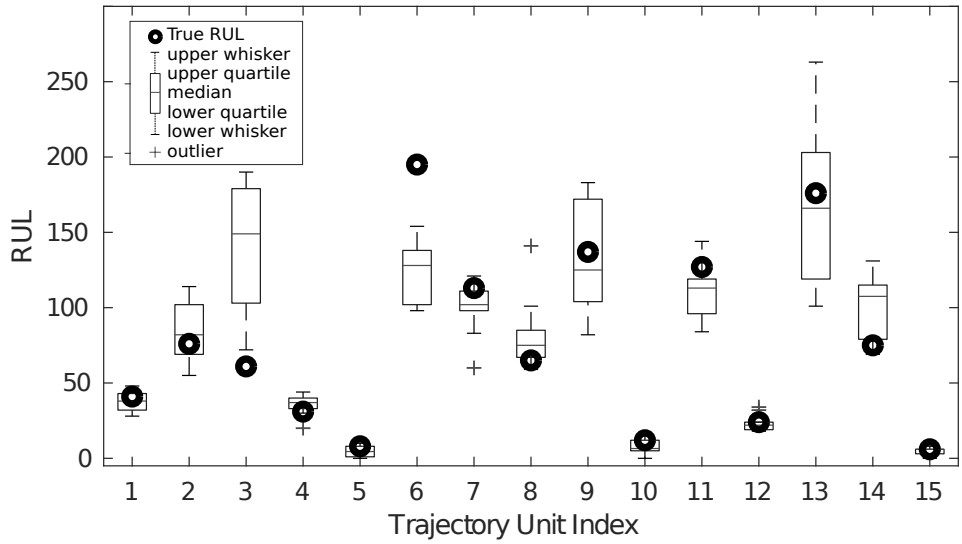

Figure 5.1: Comparison of True and Predicted RULs

Figure 5.1 provides a comparison of true and estimations RULs for the first fifteen trajectories of secondary dataset #1 (SD#1). These box plots illustrate a graphical depiction of groups of numerical RUL estimation data through their quartiles. They display the distribution of estimations based on three main outlines: lower RUL estimation quartile, median RUL estimation and upper RUL estimation quartile. The plots also have estimations extending vertically from the boxes indicating outside variabilities. The box plots demonstrate the variations in a statistical RUL prediction population without making any assumptions as to the potential underlying distributions. The change of spacing between the dimensions of box designates the degree of dispersion in the best matching RUL predictions. Most of the true RULs are

within the range of upper and lower whiskers, whereas a considerable number are actually between the upper and lower quartiles. Trajectories #3 and #6 are the only exceptions, and they are even beyond all outliers. These cases are particularly dangerous for the performance evaluation made by prognostic metrics because their high error rates are detrimental to the performance level of the entire dataset.

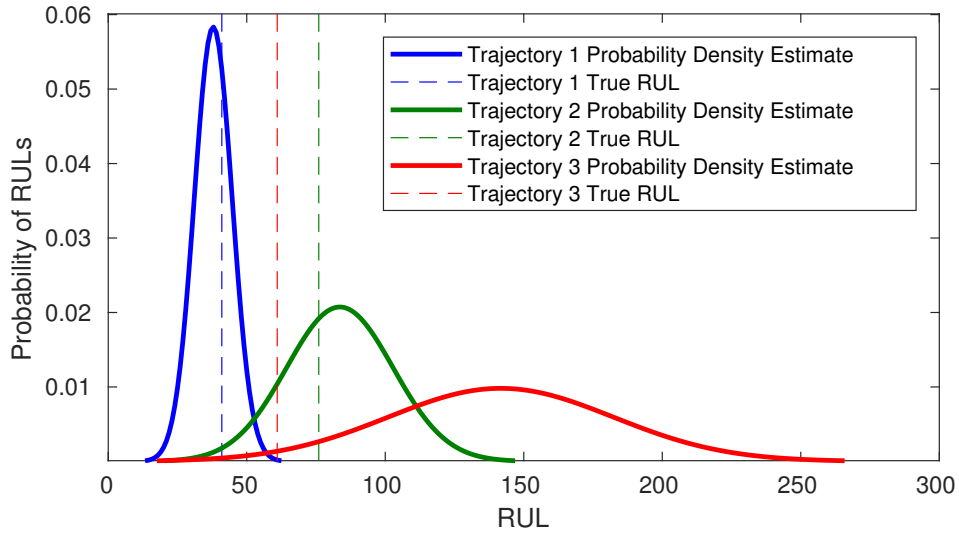

Figure 5.2: Demonstration of density estimations of predicted RULs

The estimated probability density functions of the first three trajectories of SD#1 are shown in Figure 5.2. The estimated density at any given point in the set of possible RUL values can be interpreted as a relative likelihood of remaining life. As the RUL probability is described over multiple estimations, the graph specifies RUL likelihood within a range of values, as opposed to taking on any single RUL value. It has been observed from Trajectory #3 that the likelihood of potential high error rate is wider and less peaked so that the estimations are distributed over a wide range unlike the low prediction errors. This proves that the high error rates can be initially identified and more inclusive precautions can be taken to avoid catastrophic results in terms of RUL estimation. As seen on previous box plot sample, the estimations on this

trajectory were not clustered on a particular region and the mean was not close to true RUL. Therefore, one can assume that if the RUL likelihood is within a relatively wider range, there is more risk to get higher error rates. With regards to the maintenance decision process after prognosis, the probability density estimate at these instances can be used to reduce the associated risks. As an example, the likelihood at any point through Trajectory #3, which has a high error rate, is relatively low, and the graph provides a wide range of estimations that need to be accounted for in further stages.

Table 5.2: RUL estimation range for the units with lowest performance

Trajectory Unit Index	48	421	3	382	139
True RUL	189	195	61	73	195
Predicted RUL (Mean)	96.8	99.2	141.7	163	119.2
Estimation 1	85	91	167	111	113
Estimation 2	81	87	103	178	144
Estimation 3	56	62	186	139	72
Estimation 4	106	133	119	176	115
Estimation 5	111	77	155	154	125
Estimation 6	133	117	72	196	121
Estimation 7	86	139	179	103	101
Estimation 8	127	112	190	171	85
Estimation 9	71	93	103	210	188
Estimation 10	112	81	143	192	128

Table 5.2 shows the RUL estimations of the least accurate instances in SD#1. All estimations here show that there is a wide range of RUL estimations and the probability density function of these can be used to provide more comprehensive knowledge regarding prognostic risks.

Table 5.3 extends this comparison and lays out the score function results

Table 5.3: Secondary Dataset#1 RUL estimations with lowest performance

Score	Unit	Length	A. Error	Score	Unit	Length	A. Error
1533.404	48	69	92.2	116.919	54	84	42.1
1189.408	421	65	95.8	100.026	34	32	60
623.322	3	32	69	100.026	43	113	60
297.867	382	43	57	100.026	100	51	60
217.026	139	37	75.8	100.026	105	54	60
167.282	411	43	65	95.544	28	65	44.5
166.335	337	52	58.7	94.265	107	48	40.9
163.021	6	57	71.5	85.620	328	48	58
149.405	99	32	55	84.626	125	44	40.8
133.289	68	33	49	80.450	279	83	44

for these least accurate cases. As seen from here, only a few undesired RUL estimation can overly impact the performance of the entire measures in a dataset. To avoid such undesired failures in a real world application, the distribution of the estimation can be further considered in the post prognostic actions.

As only one catastrophic failure in RUL estimation would be compelling for the validation of the proposed prognostic framework, this research uses 20 secondary dataset + 1 original data (all with 435 test trajectories) to prove that the algorithm can perform in many diverse cases. Considering that there are prognostic researches developed and tested by a single case, the proposed model holds the promise for validation with all 9.135 (21×435) different operational cases. Within all, least performance in terms of the scoring function is for Unit 48 at #SD1 and the rest demonstrate higher performance levels.

A comparison of absolute error rates and test data unit lengths for SD#1 is provided in Figure 5.3 (and also the highest rates in Table 5.3). The error rates show an exponential increase as the unit lengths decrease. Assuming that the longer test trajectories are mature enough to adequately represent system behaviour, associated RUL predictions are generally consistent and do not result in high error rates.

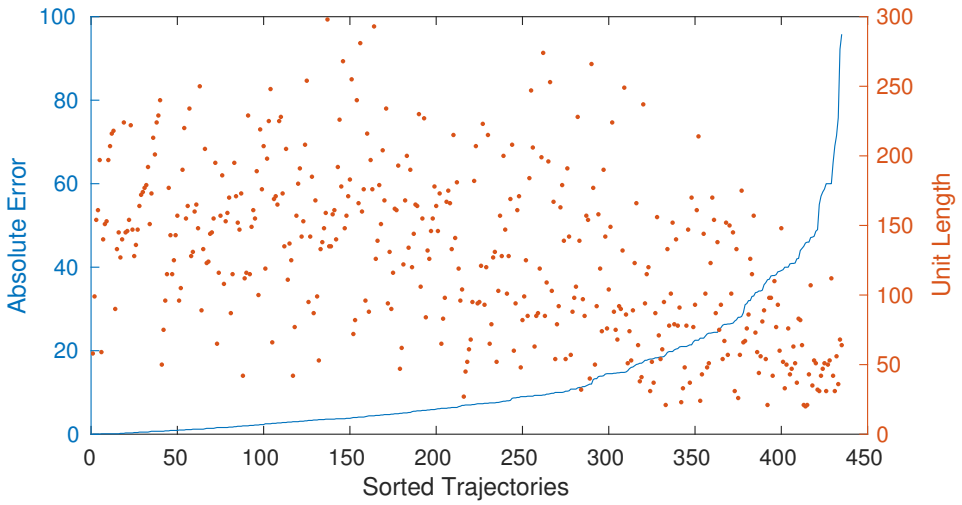

Figure 5.3: Comparison of Absolute Error and Unit Test Length for SD#1

During the case studies, it was noticed that the consistency of mature trajectories is a direct result of the grown patterns, which are not affected by undesired fluctuations in the data. For short trajectories, on the other hand, the variance in data fluctuations is a major concern as they might result in catastrophic failures in the overall accuracy of the data set.

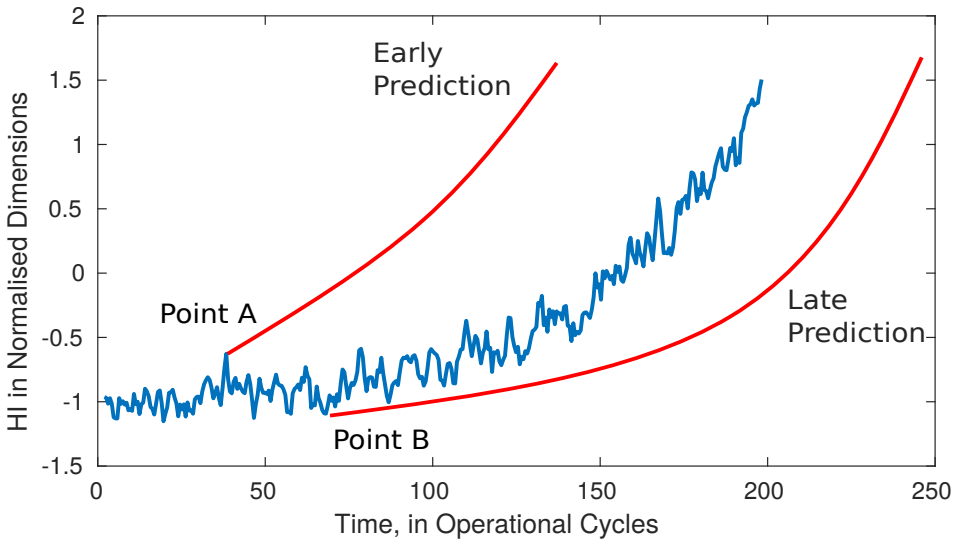

Figure 5.4: Risky positions with respect to the data fluctuation

Figure 5.4 illustrates two points within a full operational trajectory. If

the final time index of a test trajectory equals "Point A", the multi-step-ahead prediction is prone to erroneous early in time predictions due to an upper fluctuation that occurred during operation. A test trajectory terminating at "Point B" witnesses a lower fluctuation and therefore, it is prone to late in time predictions. The early and late prediction speculations for both points are demonstrated in the plot. If these predictions were applied to prognostic metrics, particularly to the scoring function, the performance of the entire data-set would, consequently, be heavily degraded.

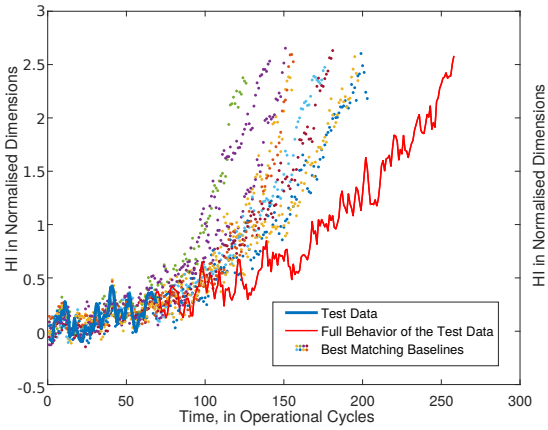

Figure 5.5: RUL calculation for Unit #421

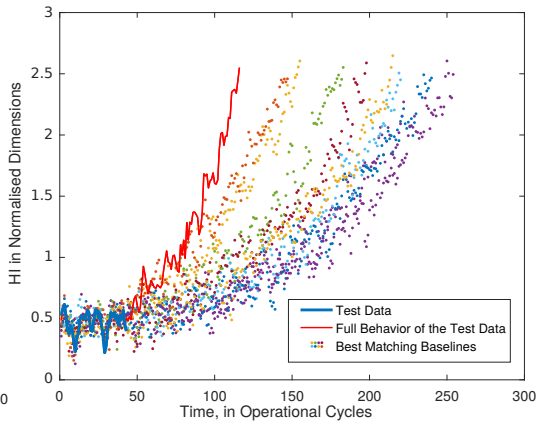

Figure 5.6: RUL calculation for Unit #3

In Figure 5.5 and Figure 5.6, the real late-in-time and early-in-time estimations for unit #421 and #3 (the ones with highest error rate in SD#1) are shown. Due to the unexpected interruption of data, the prediction cannot provide accurate RUL estimations despite the fact that the matching units still represent high levels of pairwise similarity between the test and training trajectories. However, the risk of such an unexpected interruption of data can be reduced by increasing the number of similarity based estimations since some run-to-failure training trajectories might have similar trends (fluctuations) in their earlier operational periods.

In Figure 5.7, the comparison of absolute error rate with scoring func-

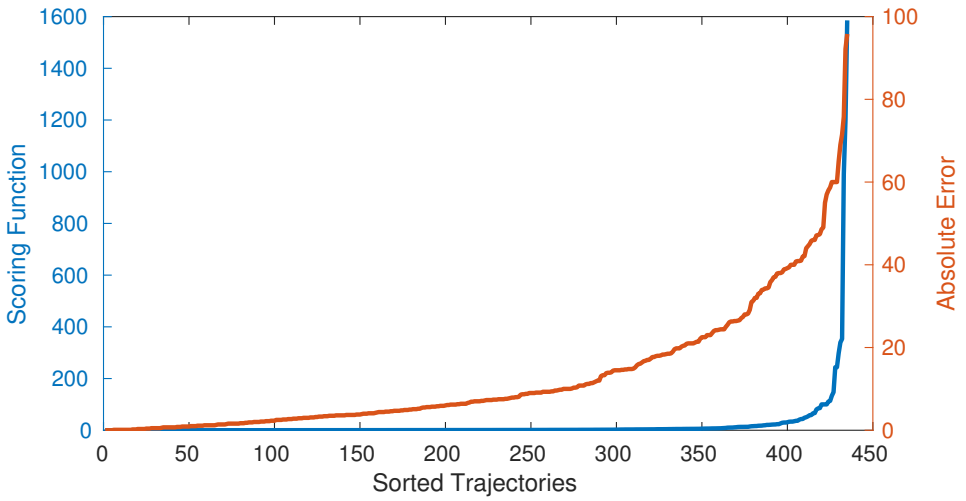

Figure 5.7: Comparison of Absolute Error and Scoring Function for Secondary Dataset #1

tion calculation for SD#1 is given. It is obvious from the graph that the relation over both metrics is asymmetrical. This is because the scoring function is calculated by an error-based exponential equation. Both plots are sorted by the number of metric measurements. Although they have a constant increase in their "y" axis, if they were sorted by a common order, the display would change due to the early and late prediction difference in the scoring function. As the prediction error rate exceeds 60 cycles, the scoring function leads to a catastrophic failure. Considering instances with high error rates and scoring function-based ranking, it can be concluded that only a few large deviations among the many predictions could noticeably escalate the metric score.

Since the well-known methods in the literature use scoring function in which the lower scores achieve higher positions, the performance evaluation of secondary data sets is initially based on this metric to benchmark the performance of presented methodology. The remaining prognostic metrics in Table 5.1 have also been applied in order to determine whether or not the designed algorithm or multi-step prediction results can show practical results. Nevertheless, these metrics are error-based measurements between a single prediction

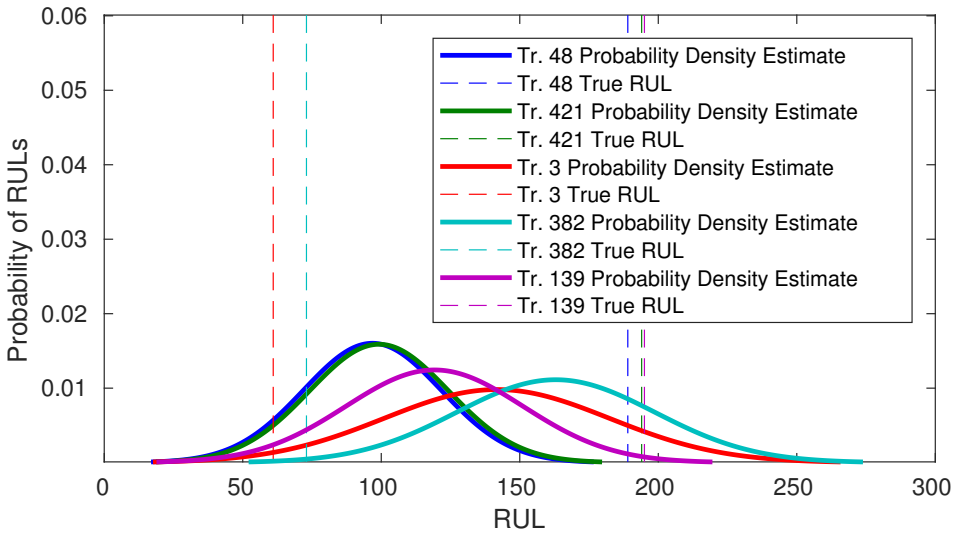

Figure 5.8: Demonstration of density estimation of predicted RULs

and a true RUL. As seen in Figure 5.8, the potential risk of high error predictions can be reduced by the short peak level of the probability index of estimations. The deviation from mean or highest probability might be relatively high for these data sets, but the true RUL calculations are generally either within lower probability rates or in a close time index region.

Table 5.4: Prognostic metrics for Secondary Data set #1 with different window sizes used for HI moving average filtering

Dataset & Window Size	Scoring Func.	FPR	FNR	MAPE	MAE	MSE
DS 1 #a	8280.807	52%	48%	24.639	13.018	410.058
DS 1 #b	7489.328	51%	49%	24.17	12.743	394.345
DS 1 #c	7396.534	49%	51%	23.552	12.398	387.982
DS 1 #mean	7298.195	50%	50%	23.685	12.541	388.8

In order to minimise the risk caused by the asymmetric characteristics of the scoring function, three different moving average window sizes for HI filtering were separately applied to the similarity-based estimations. In general, the mean of these three estimations has provided lower results for error-based prognostic metrics (see Table 5.4).

Table 5.5: Prognostic metrics for Secondary Data set #12 with different window size used for HI moving average filtering

Dataset & Window Size	Scoring Func.	FPR	FNR	MAPE	MAE	MSE
DS 12 #a	2452.792	48%	52%	24.118	11.076	260.71
DS 12 #b	2229.2	49%	51%	23.605	10.932	253.722
DS 12 #c	2137.85	47%	53%	23.27	10.878	250.864
DS 12 #mean	2123.455	49%	51%	23.4	10.86	248.391

Appendix D.1 provides an expanded version of Table 5.4 for all twenty secondary data sets. Among all these data sets, in terms of scoring function performance, dataset #1a showed the highest rate (8280.807, see Table 5.4) while dataset #12c showed the lowest rate (2123.455 see Table 5.5). Therefore, the application of the developed model to secondary dataset #12 exhibits more promising results for multi-step long-term time series predictions compared with the other secondary datasets used in case studies. This is mainly because there are not excessive error rates ($|Ti - Pi| \gg 60$) in the application of proposed model in secondary dataset #12.

Table 5.6: Secondary Dataset#12 RUL estimations with lowest performance

Score	Unit	Length	A. Error	Score	Unit	Length	A. Error
329.29	417	55	58	43.01	239	33	49.2
131.95	380	93	48.9	40.71	193	76	48.5
130.22	139	37	63.4	37.86	112	82	36.6
98.48	255	69	46	32.115	95	92	35
63.66	352	51	54.2	31.13	93	99	34.7
63.07	103	105	41.06	29.66	34	32	44.5
59.34	17	32	41	29.56	137	44	34.2
57.06	3	32	52.8	28.08	134	43	43.9
53.05	242	72	39.9	26.9	161	58	33.3
50.33	346	88	51.2	25.04	349	86	32.6

Table 5.6 demonstrates that when the lowest performance trajectories have minor error rates so that they could not affect the overall data set sig-

nificantly, the remaining scoring function level notably decreases. Therefore, it would be fair to conclude that the scoring function is based on a precision ranking in which even a few poor estimations could result in a catastrophic outcome.

The maximum absolute error rate in dataset #12 is 63.4 for test unit 139 with a length of 37 (see Table 5.6). However, the relatively lower scoring function level of 99.25 indicates that this is an early prediction rate. Maximum scoring rate, on the other hand, is 163.02 for test unit 139 with a length of 37 and an error rate of 58. These relatively lower error rates result in significantly lower scoring function estimations and higher performance rates for the overall data set. Assuming that none of the randomly selected test data locations falls within a critically high fluctuation rate, as in data set #12, the training of the network and the similarity-based prediction can accomplish learning as desired while training performance is substantially increased by collaborative data use.

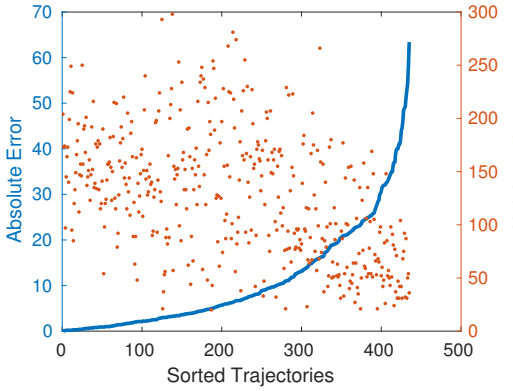

Figure 5.9: Comparison of Absolute Error and Unit Length for Secondary Dataset #12

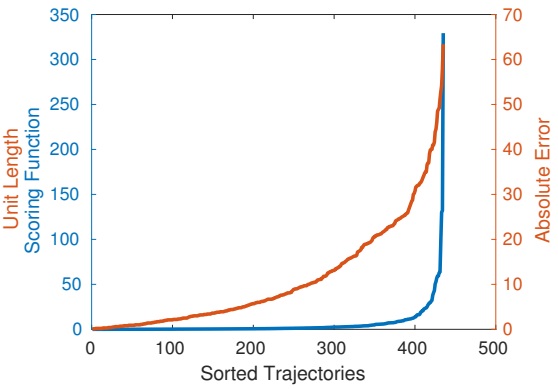

Figure 5.10: Comparison of Absolute Error and Scoring Function for Secondary Dataset #12

In Figure 5.9, the comparison of absolute error rates and unit lengths for test trajectories in SD#12 is provided. As in the case of SD#1 (Figure 5.3), there is also an exponentially increasing error rate. However, the main difference between dataset #1 and #12 is their top absolute error rates. This could

provide a less asymmetric comparison between the rates of absolute error and scoring function in SD#12 (see Figure 5.10).

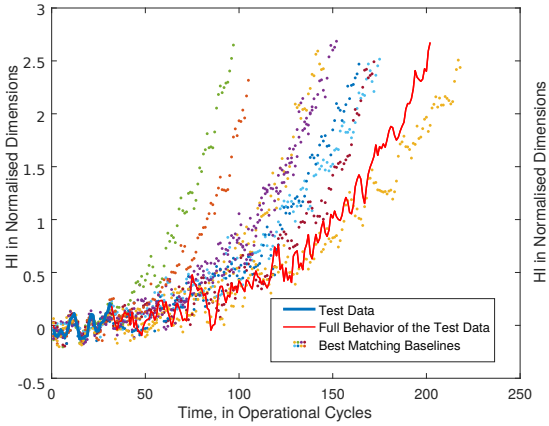

Figure 5.11: Top Matching Similarities and RUL calculation for Dataset #12, Trajectory Unit #421

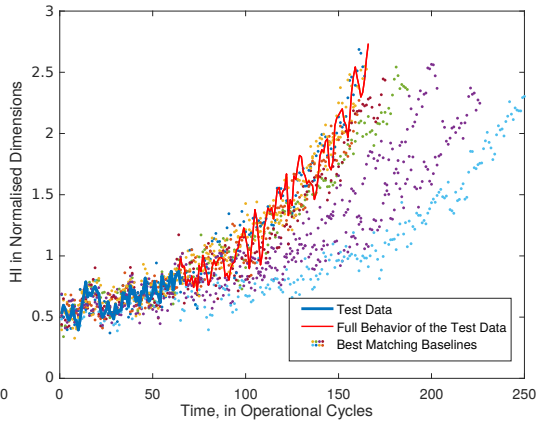

Figure 5.12: Top Matching Similarities and RUL calculation for Dataset #12, Trajectory Unit #3

Poor performance in the calculations may happen randomly because future prediction can never be absolutely estimated, especially when the dataset is very noisy, complex and ends during an abnormality. In Figure 5.11 and Figure 5.12, Trajectories #421 and #3 have resulted in more accurate estimation rates in comparison to the fluctuation inadequacy of the same trajectories in dataset #1. Despite seeming like outliers, some of the estimations in these figures appear to contribute a significant improvement in the RUL fusion. The outliers, for this reason, are considered at the RUL fusion stage, and the final RUL life is accepted simply as the mean of the top ten best matching estimations without removing the outliers.

Performance evaluation of the developed model for all secondary datasets is shown in Table 5.7. Regarding the results achieved, the quantifiable prognostic metrics assessing the performance of the datasets show promising results. It is important to note that prognostic metrics are employed to address pre-evaluation before the final test subset is sent to NASA PCoE.

Every result in Table 5.7 has a specific performance significance that

Table 5.7: Performance Evaluation of Secondary Datasets

#SD	Score	FPR	FNR	MAPE	MAE	MSE
1	7298.20	0.50	0.50	23.68	12.54	388.80
2	3104.18	0.53	0.47	23.17	11.71	312.21
3	4580.18	0.53	0.47	23.41	11.85	353.26
4	4189.98	0.52	0.48	22.46	11.14	299.88
5	3746.60	0.50	0.50	20.44	11.50	321.97
6	5779.51	0.50	0.50	24.22	12.21	354.53
7	4368.22	0.51	0.49	21.89	11.82	345.27
8	3406.43	0.46	0.54	24.29	11.50	304.79
9	5358.71	0.50	0.50	23.04	11.84	352.05
10	6523.67	0.53	0.47	22.81	11.75	366.86
11	4485.17	0.52	0.48	22.06	11.66	348.14
12	2123.46	0.49	0.51	23.40	10.86	248.39
13	5259.01	0.50	0.50	23.97	9.66	261.66
14	5238.38	0.48	0.52	24.52	12.35	336.80
15	4189.98	0.52	0.48	22.46	11.14	299.88
16	2745.90	0.51	0.49	22.75	11.23	279.71
17	3896.62	0.52	0.48	22.38	10.89	282.73
18	2189.90	0.48	0.52	21.60	10.28	236.47
19	3085.24	0.51	0.49	22.49	11.81	302.74
20	5668.86	0.44	0.56	24.42	11.60	326.57

should be monitored carefully. When any of the metrics have not been satisfied with the desired performance, the overall prediction has probably been biased by a risky effect. For example, if the scoring function is extremely high in comparison to the mean absolute error, it is likely the calculations have been biased in the direction of an early in time prediction. In a similar manner, when the false positive and negative rates deviate significantly from 50%, it can be argued that either RUL fusion is weighted in favour of a particular direction or the algorithm has an undesired tendency. The consideration of each metric is, therefore, necessary in the interests of risk reduction in terms of predictive performance.

The biases in predictions are negative inclinations to present a sided perspective in the model, often accompanied by ignoring any consideration of

the possible merits of alternative algorithm points. The algorithms may develop sided steps towards, or against individual characteristics of the datasets. In such cases, the bias could be a systematic error resulting from an unfair sampling or a prediction process that cannot provide accurate results. Therefore, the identification of biases within performance metrics is of particular importance in prognostic applications.

5.3 Benchmarking with Data Challenge

After more than 45,000 downloads since their first release in 2008 and more than 70 major publications in the literature (Ramasso and Saxena, 2014), C-MAPSS datasets have become one of the leading sources in the prognostic field. Moreover, according to PCoE (2014), the total number of applications using these data sets between the years of 2008 and 2014 was 134.

This research provides performance benchmarking for the developed model with regards to the PHM08 data challenge winning entries. The algorithm is applied to the PHM08 data challenge "final test" subset, which contains 435 test trajectories. The RUL is calculated for each test subset, and the results are sent to PCoE to be evaluated by a comprehensive set of metrics to form a standardised communication by which the developers and users demonstrate and compare their findings. These results are principally aimed at the performance validations for prognostic applications. Since they are based on true RULs and available run-to-failure data, their validation is of particular importance in the model justification where the metric feedbacks can be used to evaluate the proposed prognostic procedures.

The presented model in this research achieved a total score of **5530.12** (calculated by the " scoring function"), which is currently the overall **leading score** in the literature, as per Table 5.8 and 5.9. Since the ranking is estab-

Table 5.8: Prognostic Performance of the Presented Model

Rank	Scoring Function	MSE	FPR (%)	FNR (%)	MAPE (%)	MAE	Std. Dev.	MAD	MdAD
1	5530.12	515.35	48.51	50.80	20.68	15.93	1.07	16.63	10.20

Table 5.9: Previous Leader board of the "final data set" of PHM08 (Ramasso and Saxena, 2014)

Rank	Scoring Function	MSE	FPR (%)	FNR (%)	MAPE (%)	MAE	Std. Dev.	MAD	MdAD
1	5636.06	546.60	64.83	31.72	19.99	16.23	1.01	16.33	11.00
2	6691.86	560.12	63.68	36.32	17.92	15.38	1.03	16.29	8.08
3	8637.57	672.17	61.38	23.45	20.72	17.69	1.09	17.79	11.00
4	9530.35	741.20	58.39	39.54	34.93	20.19	1.22	20.17	15.00
5	10571.58	764.82	58.85	41.15	32.60	20.05	1.22	20.41	14.23
6	11572*	-	-	-	-	-	-	-	-
7	14275.60	716.65	59.77	37.01	21.61	18.16	1.17	18.57	11.00
8	19148.88	822.06	56.09	41.84	30.25	20.23	1.29	20.89	13.00
9	20471.33	1000.06	51.95	48.05	33.63	22.44	1.42	24.05	14.78
10	22755.85	1078.19	62.53	35.40	39.90	24.51	1.45	24.08	20.00
11	25921.26	854.57	34.25	64.83	51.38	22.66	1.36	21.49	16.00

* After 2008

lished based on the PHM Scoring function, there is not a penalty limit for predictions, and thereby a high error rate among all predictions in the dataset could affect the ranking of the model.

The lack of limitation is particularly risky for test units in which the available condition monitoring history is short, and both early and late predictions are likely to happen. In order to avoid the risk of excessive error punishment in terms of scoring function, the maximum threshold value for RUL predictions is directly adjusted to be 135. As the data is explored, it is found that the maximum RUL is 195 in the data sets where true RUL is provided (FD001 to FD004 which are simulated in operational cycles similar to PHM08 challenge data). Even in the worst scenarios of maximum RULs, the score that is adjusted to 135 would be $\sim 100 = exp(\frac{\sim 195-135}{13})$. However, in an undesired event of late prediction for a short true RUL (say those of $T = 100$

and $P = 185$), the unbounded prediction might result in catastrophic results, $\sim 4915 = exp(-\frac{100-\sim 185}{10})$, rather than having much less punishment with bounded RUL, $\sim 33 = exp(-\frac{100-135}{10})$. Different threshold values are tested by the secondary data sets and the value "135" is found to be optimal. This number is only for limitation of high punishment and its exact effect cannot be known since the true RULs are not provided to the developers. However, it is observed that it can reduce the scoring function outcomes for most of the secondary datasets.

According to the performance results reported in Table 5.9, the multi step forecast performance over the long-term cycles appeared satisfactory as the metric results are balanced. The developed model seems to exhibit promising results for multi-step long-term time series predictions for exponential wear growth. The training of the network could accomplish learning as desired while training performance is substantially increased by using different training data trajectories. With the use of collaborative RUL prediction, there appears to be a significant improvement in error-based metric measurements. The RUL fusion can be effectively received by applying multiple training sets within a specific test subset. It is important to note that some calculations might have resulted in high error rates, but the performance of the overall algorithm still results in an expected range with regards to the previous secondary dataset findings.

One further point in Table 5.9 that deserves a special consideration for final test validation is the distinguishing difference between False Positive (FP) and False Negative (FN) rates. These percentages and their relation to each other are descriptive measurements for assessing the consistency and reliability of developed models. As is evident from the previous challenge leader board, the negative rates are predominantly lower than the positive rates. This can be attributed to the algorithm's intentional weighted RUL selection because of the

asymmetric high penalty scores for late in time predictions. This research did not apply any weighting method because such an approach would be unrealistic in real life scenarios and it is also observed in the secondary datasets that the early in time predictions might also dominate the final score. That's why, the obtained FP and FN rates are close to each other, and the proposed framework can be regarded as sustained and reliable in terms of these rates.

5.4 Summary

This chapter brings forward the performance evaluation, analysis and validation of the presented prognostic methodology, as based on the case studies presented in the previous chapter. The chapter also deals with the issues of critical assessment of how prognostic performance can be improved by using the information from secondary datasets as well as including true RULs into the validation.

A generic set of prognostic metrics are employed to form a standardised communication and benchmarking with the literature. This allows the prognostic algorithm to be validated with existing methods, and the minor stages in its framework (such as using different windows size in neural network estimation filtering or bounding the RUL to a maximum threshold) can be reconfigured when there is an evidence of higher performance. The proposed methodology and its application in secondary datasets prove its capability through the prognostic performance metrics. However, when the literature is considered, the performance analysis needs to be compared with other similar studies and the results should be evaluated by a third party rather than any party involved in the implementation.

In the cases where the true RULs are available, the test data sets allow a trial-and-error type of validation, and therefore the methodologies might be

biased in favour of the dataset. Such a misapprehension surrounding prognostic actions would limit the application of the model to other scenarios. To overcome these issues, the results from PHM08 data challenge "final test" subset are sent to NASA PCoE to be validated by prognostic metrics and the outcomes are compared with the winning entries in challenge leaderboard. The metric results received from PCoE proves that the proposed method could achieve the current overall top score recorded since the first release of dataset in 2008.

Chapter 6

Conclusions and Future Work

This chapter summarises the major contributions in the proposed methodology and provides several comments for future research. The main motivation behind this research is to design an adaptive data filtering model as a part of multi-step ahead RUL estimations for complex systems working under various regimes and operational conditions. A prognostic framework for multidimensional and multivariate data is developed in an attempt to enhance the RUL estimation accuracy and model robustness. This work not only introduce a contribution to the concept of complex system prognostics, but also manifests a hybrid dimension reduction model for condition monitoring data, standardisation of multiple regimes into a common domain even for the upcoming novel trajectories which are not found in the original condition monitoring data ,a similarity-based estimation approached that is designed to make collaborative RUL calculations owing to the data filtering model, and a number of modifications to improve the performance of the prognostic approaches in terms of prognostics metrics. Additionally, the thesis introduces a comprehensive literature review on the progress of the current prognostic applications as well as the various sources and methods applied in complex system domains.

6.1 Contributions

Selecting prognostic algorithms for a particular domain is crucial to the success of a RUL estimation framework. Each prognostic method has specific requirements to capture the degradation process and has its own merits and limitations. The research developed a multifaceted prognostic model consisting of data processing and multi-step ahead predictions to integrate any desired advantages whilst avoiding characteristic disadvantages. To that end, a hybrid prognostic framework of three advanced approaches (multi regime normalisation, feed-forward neural network and similarity-based prediction) is proposed in the methodology section to overcome the issues presented by the multidimensional condition monitoring of complex systems under dynamic operating regimes.

In terms of the contributions of this hybrid model, this research proposed a novel feature extraction model to normalise the entire data population (and also upcoming novel data) with regard to the characteristics of the operational cases. This mode of the proposed methodology can preserve operational characteristics and standardise health degradation into a common scale. This is mainly crucial for demonstrating the strengths of collaboration in RUL predictions that can be made through the similarities between existing or potential novel operational cases.

For the case studies, to analyse the integration of the methodology, the proposed framework is considered in a gas turbine engine degradation scenario. "PHM08 Challenge Data Set" is used to overcome the prognostic problems on common datasets and mutual comparisons. Additionally, the framework is tested with the synthetic HI adaptation model, re-generated raw data and twenty different secondary datasets to provide increased knowledge that could feasibly allow alternative applications.

6.2 Comments

The application of suggested techniques shows that the designed algorithm was able to detect an unknown RUL effectively and could predict the exponential wear level of the system in multi-step long-term cycles. The deficiencies exhibited by data restrictions and scattered operating conditions are removed through neural network data filtering. The application of network filtering in trajectory similarity-based multi-step-ahead estimation model has provided a collaborative RUL estimation model, which resulted in significant performance results. As the accuracy and understanding of data pre-processing have increased, RUL forecast and future wear level predictions could also acquire higher performance levels.

Regarding the research objectives defined within the Introduction chapter, the conclusion can be drawn that the literature survey, methodology and case studies of the thesis show an effective adaptive data filtering as a part of multi-step ahead RUL estimation. The research first introduces an overall understanding of emerging concepts in prognostics and analyses the potential use of different prognostic methods in the maintenance of complex systems. According to the merits and limitations of the current literature, a hybrid prognostic model is introduced as a combination of feature extraction and prediction modes. To that end, a data filtering processes is designed to model a training library in which health of various operational cases (both existing or novel ones) can be calculated separately. This allows to form a merging process between qualification of raw data and life estimations. Consequently, the results prove that the prediction performance is increased, excessive error rates for critical cases are reduced and the gap between false positive and negative RUL error rates are minimised.

The prognostic algorithm introduced in the methodology chapter has been effectively applied in complex system data sets and it could fill the re-

search gap that motivated this study by acquiring the training data from a larger group of suppliers and applying a collaborative estimation model. A hybrid data-driven filtering method could successfully deliver the desired health indicator targets into similarity-based RUL estimations. As the multi regime normalisation algorithm is applied with the trained neural network, there is no longer a need to normalise the entire data set at once. On account of this, both test and training trajectories can be processed individually by maintaining their characteristics.

6.3 Implications and Limitations

The presented prognostic framework in the methodology chapter proves to be promising to be applied in a engineering systems under dynamic operational regimes. In the case studies, it is observed that the model can detect and standardise the failure degradation in multiple domains at any time index of the operational cycles. The main prognostic problem is to estimate the RULs of complex systems along with normalising their failure degradation from different regimes into a single domain and also preserving the characteristics of operational cases with regard to each other a potential novel cases. The proposed methodology could overcome these issues and provide filtered HIs from a network library for a collaborative similarity based RUL estimation.

During the trial of methodology with synthetic data, it has been revealed that a mathematically defined synthetic target could also perform adequately in the proposed Bayesian regularised feed forward neural network training. In consequence of the concerns regarding the filtering deficiencies, scoring function, one-time limited submission for validation and variance in initial wear levels, a step-by-step HI (output) data processing is preferred. There is, however, a considerable potential for the developed network model

to automate output selection and the normalisation of various failure mode situations.

6.4 Further Work

The work carried out in this thesis has revealed many promising areas of further work in prognostics field and the proposed framework prove to be promising. A few of these areas worthy of potential research areas can assist for the foreseeable future of prognostic research. The following suggestions provide pointers towards a further research for future works.

As described in section 2.3.4, a prognostic strategy may perform less well than expected when applied in different scenarios and an alternative methodology can outperform when it concentrates on the characteristics of a specific domain. Therefore, an effective prognostic programme does not necessarily rely on a particular algorithm and there are many other potential factors playing a significant role in enhancing prognostic performance such as recognising an inadequate feature, identifying the fault location, determining inadequate feature and the fault mode and estimating when a failure will happen. However, when the potential of the presented network model of automatic output selection and normalisation, the prognostic programme can be designed in a way to allow generalisation into broader domains.

The key areas for further research will include the investigation of the applicability of automatic output assignment. A smart target selection system can be formed of a library of synthetic output models via various mathematical models and selecting the best matching degradation model for the raw data. For future work, a more advanced smart filtering model based on the presented neural network algorithm can be applied to various prognostic applications. As observed in section 4.5 (Figure 4.26), the proposed network function can effec-

tively filter the data when it is designed with a synthetic target. Even though the synthetic data cannot precisely demonstrate the degradation behaviour and has a constant exponential increase through the operational time units rather than having a constant linear increase in the initial stages, the network estimation can identify a well-defined pattern and correct the misbehaviour of training target. Therefore, one can argue that the target assignment for network training can be automated in a way that the function performs self-acting (or correcting) behaviour. Such an automated function would allow for quick understanding of condition monitoring data to aid in RUL estimation. It would also allow for more complex algorithm protocols that are impractical using manual modelling.

Several improvements can also be made on the RUL estimation mechanism such as adapting the similarity based prediction with the neural network structure within the future failure mode. Along with a smart filtering model, a further adaptation on the multi-step-ahead prediction for complex systems will notably contribute to a potential development in methodology.

In the results (chapter 5, section 5.2), it is observed that the cases with high prediction error are distributed over a wide range unlike the low prediction errors. Thereby, the probability density at any given point of a high prediction error is wider and less peaked. Since the prognostic metrics in the literature are based on the error between a single RUL estimation and actual time to failure, they cannot properly evaluate these cases. Therefore, the further work can be extended to tailor the prognostic metrics with an attempt to provide more functional and practical assessments for post maintenance actions.

Some further research can also be expected to explore additional secondary data collection mechanism by integrating it to the framework as a loop allowing the method to find potential risky cases and the potential solutions. These datasets can serve a purpose as a benchmark resource for prognostic

algorithms to be tested on. The empirical prognostic frameworks suffer from the lack of the available condition monitoring data, and therefore, a potential further research area to elicit prognostics development is dissemination of common data sets. Considering the industrys confidentially issues and shortage of available data sets, the secondary data reconstruction can give clear degradation information and aid developers to improve their algorithms. Additionally, the secondary data can be used as an internal feed back source from which the algorithm can understand the potential risks and alternative degradation patterns.

These potential research areas for future work can present an improvement to proposed prognostic model and an automated method for identifying appropriate prognostic modes from available data sources. Additionally, the development of a self-acting methodology can be discussed for further prognostic applications. Such a method, as well as the prognostic algorithms in this research, will allow to aid in rapid prognostic model prototyping and full health degradation feature extraction.

References

Abbas, M. (2010). *System-level health assessment of complex engineered processes*. PhD thesis, Georgia Institute of Technology.

Ahmad, R. and Kamaruddin, S. (2012). An overview of time-based and condition-based maintenance in industrial application. *Computers & Industrial Engineering*, 63(1):135–149.

Aksoy, S. and Haralick, R. M. (2001). Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern recognition letters*, 22(5):563–582.

An, D., Choi, J.-H., and Kim, N. H. (2013). Prognostics 101: A tutorial for particle filter-based prognostics algorithm using matlab. *Reliability Engineering & System Safety*, 115:161–169.

Anderson, T. W. (2011). *The statistical analysis of time series*, volume 19. John Wiley & Sons.

Badıa, F., Berrade, M. D., and Campos, C. A. (2002). Optimal inspection and preventive maintenance of units with revealed and unrevealed failures. *Reliability Engineering & System Safety*, 78(2):157–163.

Bar-Yam, Y. (2003). Complexity of military conflict: Multiscale complex systems analysis of littoral warfare. *Report to Chief of Naval Operations Strategic Studies Group*.

Barad, S. G., PV, R., et al. (2012). Neural network approach for a combined performance and mechanical health monitoring of a gas turbine engine. *Mechanical Systems and Signal Processing*, 27:729–742.

Baraldi, P., Compare, M., Sauco, S., and Zio, E. (2013). Ensemble neural network-based particle filtering for prognostics. *Mechanical Systems and Signal Processing*, 41(1):288–300.

Barlow, R. E. and Proschan, F. (1996). *Mathematical theory of reliability*. Society for Industrial and Applied Mathematics.

Barton, P. H. (2007). Prognostics for combat systems of the future. *IEEE Instrumentation & Measurement Magazine*, 10(4):10–14.

Baruah, P. and Chinnam, R. B. (2005). Hmms for diagnostics and prognostics in machining processes. *International Journal of Production Research*, 43(6):1275–1293.

Bask, A., Spens, K., Uusipaavalniemi, S., and Juga, J. (2008). Information integration in maintenance services. *International Journal of Productivity and Performance Management*, 58(1):92–110.

Batko, W. (1984). Prediction method in technical diagnostics. *Unpublished doctoral dissertation, Cracov Mining Academy*.

Baurle, R. A. and Gaffney, R. L. (2008). Extraction of one-dimensional flow properties from multidimensional data sets. *Journal of Propulsion and Power*, 24(4):704–714.

Bayes, T., Price, R., and Canton, J. (1763). *An essay towards solving a problem in the doctrine of chances*. C. Davis, Printer to the Royal Society of London.

Bazovsky, I. (1961). *Reliability Mathematics: Fundamentals*. McGraw-Hill Book Co. New York.

Beale, M. and Demuth, H. (1998). Neural network toolbox. *For Use with MATLAB, User's Guide, The Math Works, Natick*, pages 1–6.

Bektas, O. (2014). A framework for performance metrics of rotating machinery prognostics within maintenance free operation period. In *Poster Presentation, WMG Research and Innovation Conference*.

Bektas, O. (2015). Narx model in gas turbine prognostics. In *Presentation, WMG Doctoral Research and Innovation Conference*.

Bektas, O., Alfudail, A., and Jones, J. A. (2017). Reducing dimensionality of multi-regime data for failure prognostics. *Journal of Failure Analysis and Prevention*, 17(6):1268–1275.

Bektas, O. and Jones, J. A. (2015). A degradation prognostic framework for gas turbine engines. In *The Twelfth International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*.

Bektas, O. and Jones, J. A. (2016). Narx time series model for remaining useful life estimation of gas turbine engines. In *Third European Conference of the Prognostics and Health Management Society*.

Bengtsson, M. (2004). Condition based maintenance systems: an investigation of technical constituents and organizational aspects. Master's thesis, Mälardalen University.

Biagetti, T. and Sciubba, E. (2004). Automatic diagnostics and prognostics of energy conversion processes via knowledge-based systems. *Energy*, 29(12-15):2553–2572.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

Bohlin, M., Doganay, K., Kreuger, P., Steinert, R., and Warja, M. (2010). Searching for gas turbine maintenance schedules. *AI Magazine*, 31(1):21–36.

Bonissone, P. P. and Goebel, K. (2002). When will it break? a hybrid soft computing model to predict time-to-break margins in paper machines. In *International Symposium on Optical Science and Technology*, pages 53–64. International Society for Optics and Photonics.

Boussif, A. (2016). *Contributions to Model-based Diagnosis of Discrete-Event Systems*. PhD thesis, Université de Lille1-Sciences et Technologies.

Box, G. E., Jenkins, G. M., and Reinsel, G. C. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Brotherton, T., Grabill, P., Wroblewski, D., Friend, R., Sotomayer, B., and Berry, J. (2002). A testbed for data fusion for engine diagnostics and prognostics. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 6, pages 6–6. IEEE.

Brotherton, T., Jahns, G., Jacobs, J., and Wroblewski, D. (2000). Prognosis of faults in gas turbine engines. In *Aerospace Conference Proceedings, 2000 IEEE*, volume 6, pages 163–171. IEEE.

Brown, P. and Sondalini, M. (2014). The evolution of maintenance practices. *Last checked, July 2016, from (http://www.lifetime-reliability.com/free-articles/maintenance-management/Evolution_of_Maintenance_Practices.pdf)*.

BSI (1993). Glossary of terms used in terotechnology bs 3811: 1993. *British Standards Institution Publishing Limited, London ISBN*, 58022484(8).

Bui, D. T., Pradhan, B., Lofman, O., Revhaug, I., and Dick, O. B. (2012). Landslide susceptibility assessment in the hoa binh province of vietnam: a comparison of the levenberg–marquardt and bayesian regularized neural networks. *Geomorphology*, 171:12–29.

Bunks, C., McCarthy, D., and Al-Ani, T. (2000). Condition-based maintenance of machines using hidden markov models. *Mechanical Systems and Signal Processing*, 14(4):597–612.

Burden, F. and Winkler, D. (2009). Bayesian regularization of neural networks. *Artificial Neural Networks: Methods and Applications*, pages 23–42.

Butler, K. L. (1996). An expert system based framework for an incipient failure detection and predictive maintenance system. In *Intelligent Systems Applications to Power Systems, 1996. Proceedings, ISAP'96., International Conference on*, pages 321–326. IEEE.

Butler, S. (2012). *Prognostic Algorithms for Condition Monitoring and Remaining Useful Life Estimation*. PhD thesis, National University of Ireland.

Byington, C. S., Roemer, M. J., and Galie, T. (2002). Prognostic enhancements to diagnostic systems for improved condition-based maintenance [military aircraft]. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 6, pages 6–6. IEEE.

Byington, C. S., Watson, M., and Edwards, D. (2004a). Data-driven neural network methodology to remaining life predictions for aircraft actuator components. In *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, volume 6, pages 3581–3589. IEEE.

Byington, C. S., Watson, M., Edwards, D., and Stoelting, P. (2004b). A model-based approach to prognostics and health management for flight con-

trol actuators. In *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, volume 6, pages 3551–3562. IEEE.

Byington, C. S., Watson, M. J., and Bharadwaj, S. P. (2008). Automated health management for gas turbine engine accessory system components. In *Aerospace Conference, 2008 IEEE*, pages 1–12. IEEE.

Camci, F. (2005). *Process monitoring, diagnostics and prognostics using support vector machines and hidden Markov models*. PhD thesis, Wayne State University, Detroit.

Cattin, D. P. (2013). Image restoration: Introduction to signal and image processing. *MIAC, University of Basel. Retrieved*, 11:93.

Celaya, J., Wysocki, P., Goebel, K., Wysocki, P., and Goebel, K. (2009). Igbt accelerated aging data set. *NASA Ames Prognostics Data Repository*.

Cempel, C. (1987). Simple condition forecasting techniques in vibroacoustical diagnostics. *Mechanical Systems and Signal Processing*, 1(1):75–82.

Cempel, C. (2003). Multidimensional condition monitoring of mechanical systems in operation. *Mechanical systems and signal processing*, 17(6):1291–1303.

Cempel, C. (2009). Generalized singular value decomposition in multidimensional condition monitoring of machines—a proposal of comparative diagnostics. *Mechanical Systems and Signal Processing*, 23(3):701–711.

Chaovalit, P. and Zhou, L. (2005). Movie review mining: A comparison between supervised and unsupervised classification approaches. In *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, pages 112c–112c. IEEE.

Chatterjee, S. and Hadi, A. S. (1986). Influential observations, high leverage points, and outliers in linear regression. *Statistical Science*, pages 379–393.

Chen, Z., Yang, Y., and Hu, Z. (2012). A technical framework and roadmap of embedded diagnostics and prognostics for complex mechanical systems in prognostics and health management systems. *IEEE Transactions on Reliability*, 61(2):314–322.

Coble, J. and Hines, J. W. (2009). Identifying optimal prognostic parameters from data: a genetic algorithms approach. In *Annual conference of the prognostics and health management society, san diego, ca, september*.

Coble, J. and Hines, J. W. (2011). Applying the general path model to estimation of remaining useful life. In *International Journal of Prognostics and Health Management*.

Coble, J. B. (2010). *Merging data sources to predict remaining useful life–an automated method to identify prognostic parameters*. PhD thesis, University of Tennessee.

Cross, S. S., Harrison, R. F., and Kennedy, R. L. (1995). Introduction to neural networks. *The Lancet*, 346(8982):1075–1079.

Daigle, M. J., Roychoudhury, I., Biswas, G., Koutsoukos, X. D., Patterson-Hine, A., and Poll, S. (2010). A comprehensive diagnosis methodology for complex hybrid systems: A case study on spacecraft power distribution systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(5):917–931.

De Almeida, R. G. T., da Silva Vicente, S. A., and Padovese, L. R. (2002). New technique for evaluation of global vibration levels in rolling bearings. *Shock and Vibration*, 9(4, 5):225–234.

Demuth, H., Beale, M., and Hagan, M. (2008). Neural network toolbox™ 6. *User's guide*, pages 37–55.

Demuth, H., Beale, M., and Hagan, M. (2015). Matlab: Neural network toolbox: User's guide matlab r2015b. *The MathWorks2009*.

Diallo, O. N. (2010). *A data analytics approach to gas turbine prognostics and health management*. PhD thesis, Georgia Institute of Technology.

Dmitry, K. and Dmitry, V. (2004). An algorithm for rule generation in fuzzy expert systems. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 212–215. IEEE.

Dong, M. and He, D. (2007). A segmental hidden semi-markov model (hsmm)-based diagnostics and prognostics framework and methodology. *Mechanical Systems and Signal Processing*, 21(5):2248–2266.

Ede, K., Ogbonnaya, E., Lilly, M., Ogaji, S., and Probert, S. (2010). Vibration monitoring of rotating systems. *Engineering*, 2(1):46.

Efthymiou, K., Papakostas, N., Mourtzis, D., and Chryssolouris, G. (2012). On a predictive maintenance platform for production systems. *Procedia CIRP*, 3:221–226.

Eker, Ö. F. (2015). *A Hybrid Prognostic Methodology and its Application to Well-Controlled Engineering Systems*. PhD thesis, Cranfield University.

Eker, Ö. F., Camci, F., and Jennions, I. K. (2012). Major challenges in prognostics: study on benchmarking prognostic datasets. *European Conference of Prognostics and Health Management Society 2012*.

Eker, Ö. F., Camci, F., and Jennions, I. K. (2014). A similarity-based prognostics approach for remaining useful life prediction. *European Conference of Prognostics and Health Management Society 2014*.

Eklund, N. H. (2006). Using synthetic data to train an accurate real-world fault detection system. In *Computational Engineering in Systems Applications, IMACS Multiconference on*, volume 1, pages 483–488. IEEE.

Elasha, F., Greaves, M., Mba, D., and Addali, A. (2015a). Application of acoustic emission in diagnostic of bearing faults within a helicopter gearbox. *Procedia CIRP*, 38:30–36.

Elasha, F., Mba, D., and Ruiz-Carcel, C. (2016). A comparative study of adaptive filters in detecting a naturally degraded bearing within a gearbox. *Case Studies in Mechanical Systems and Signal Processing*, 3:1–8.

Elasha, F., Mba, D., and Teixeira, J. A. (2015b). Gearboxes prognostics with application to tidal turbines. In *Proceedings of twelfth international conference on condition monitoring and machinery failure prevention technologies. Oxford, UK.*

Elasha, F., Mba, D., Teixeira, J. A., and Cranfield, B. (2014a). Failure prediction of tidal turbines gearboxes. *eMaintenance*, page 49.

Elasha, F., Mba, D., Teixeira, J. A., and Togneri, M. (2015c). Life prediction of tidal turbine gearboxes. In *EWTEC conference, Nantes, France*, pages 7–10.

Elasha, F., Mba, D., and Teixeria, J. A. (2014b). Condition monitoring philosophy for tidal turbines. *International Journal of Performability Engineering*, 10(5).

Elasha, F., Mba, D., Togneri, M., Masters, I., and Teixeira, J. A. (2017). A hybrid prognostic methodology for tidal turbine gearboxes. *Renewable Energy*, 114:1051–1061.

Elasha, F., Ruiz-Carcel, C., Mba, D., and Chandra, P. (2014c). A comparative study of the effectiveness of adaptive filter algorithms, spectral kurtosis and linear prediction in detection of a naturally degraded bearing in a gearbox. *Journal of Failure Analysis and Prevention*, 14(5):623.

Elasha, F., Ruiz-Cárcel, C., Mba, D., Kiat, G., Nze, I., and Yebra, G. (2014d). Pitting detection in worm gearboxes with vibration analysis. *Engineering Failure Analysis*, 42:366–376.

Elattar, H. M., Elminir, H. K., and Riad, A. (2016). Prognostics: a literature review. *Complex & Intelligent Systems*, 2(2):125–154.

Elforjani, M. (2016). Estimation of remaining useful life of slow speed bearings using acoustic emission signals. *Journal of Nondestructive Evaluation*, 35(4):62.

Ellis, B. A. and Byron, A. (2008). Condition based maintenance. *The Jethro Project*, 10:1–5.

Feng, E., Yang, H., and Rao, M. (1998). Fuzzy expert system for real-time process condition monitoring and incident prevention. *Expert Systems with Applications*, 15(3-4):383–390.

Fink, O., Zio, E., and Weidmann, U. (2014). Predicting component reliability and level of degradation with complex-valued neural networks. *Reliability Engineering & System Safety*, 121:198–206.

Foresee, F. D. (1996). *Generalization and neural networks*. PhD thesis, Oklahoma State University.

Foresee, F. D. and Hagan, M. T. (1997). Gauss-newton approximation to bayesian learning. In *Neural Networks, 1997., International Conference on*, volume 3, pages 1930–1935. IEEE.

Frederick, D., DeCastro, J., and Litt, J. (2007). User's guide for the commercial modular aero-propulsion system simulation (c-mapss)(tech. rep.). *Cleveland, Ohio*, 44135.

Freedman, D. A. (2009). *Statistical models: theory and practice*. cambridge university press.

Galati, F. A., Forrester, D., and Dey, S. (2008). Application of the generalised likelihood ratio algorithm to the detection of a bearing fault in a helicopter transmission. *Australian Journal of Mechanical Engineering*, 5(2):169–176.

Garga, A., McClintic, K., Campbell, R., Yang, C.-C., Lebold, M., Hay, T., and Byington, C. (2001). Hybrid reasoning for prognostic learning in cbm systems. In *Aerospace Conference, 2001, IEEE Proceedings.*, volume 6, pages 2957–2969. IEEE.

George, V., Frank, L., Michael, R., Andrew, H., and Biqing, W. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. John Wiley & Sons, Inc.

Ghahramani, Z. (2001). An introduction to hidden markov models and bayesian networks. *International journal of pattern recognition and artificial intelligence*, 15(01):9–42.

Goebel, K., Qiu, H., Eklund, N., and Yan, W. (2007). Modeling propagation of gas path damage. In *Aerospace Conference, 2007 IEEE*, pages 1–8. IEEE.

Goebel, K., Saha, B., Saxena, A., Celaya, J. R., and Christophersen, J. P. (2008a). Prognostics in battery health management. *IEEE instrumentation & measurement magazine*, 11(4).

Goebel, K., Saha, B., Saxena, A., Mct, N., and Riacs, N. (2008b). A comparison of three data-driven techniques for prognostics. In *62nd Meeting of*

the Society For Machinery Failure Prevention Technology (MFPT), pages 119–131.

Goebel, K., Saxena, A., Saha, S., Saha, B., and Celaya, J. (2011a). Prognostic performance metrics. *Machine Learning and Knowledge Discovery for Engineering Systems Health Management*, 147.

Goebel, K., Vachtsevanos, G., and Orchard, M. E. (2013). *Prognostics*. NASA, Prognostics - Intelligent Systems Division.

Goebel, K. F., Eklund, N. H. W., Qiu, H., and Yan, W. (2011b). System and method for damage propagation estimation. US Patent 7,933,754.

Goode, K., Moore, J., and Roylance, B. (2000). Plant machinery working life prediction method utilizing reliability and condition-monitoring data. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, 214(2):109–122.

GPA (2013). Gpal performance monitoring and diagnostics system. *Last checked, July 2014, frm (http://www.gpal.co.uk/)*.

Greitzer, F. L., Kangas, L. J., Terrones, K. M., Maynard, M. A., Wilson, B. W., Pawlowski, R. A., Sisk, D. R., and Brown, N. B. (1999). Gas turbine engine health monitoring and prognostics. In *International Society of Logistics (SOLE) 1999 Symposium, Las Vegas, Nevada*, volume 30, pages 1–7.

Groenevelt, H., Pintelon, L., and Seidmann, A. (1992). Production batching with machine breakdowns and safety stocks. *Operations research*, 40(5):959–971.

Gulati, S. and Mackey, R. (2003). Prognostics methodology for complex systems. *NASA Tech Briefs*.

Günel, A., Meshram, A., Bley, T., Schuetze, A., and Klusch, M. (2013). Statistical and semantic multisensor data evaluation for fluid condition monitoring in wind turbines. In *Proc. 16th Intl. Conf. on Sensors and Measurement Technology, Germany*.

Guo, Q., Wu, W., Massart, D., Boucon, C., and De Jong, S. (2002). Feature selection in principal component analysis of analytical data. *Chemometrics and Intelligent Laboratory Systems*, 61(1):123–132.

Hagan, M. T. and Demuth, H. B. (1999). Neural networks for control. In *American Control Conference, 1999. Proceedings of the 1999*, volume 3, pages 1642–1656. IEEE.

Hagan, M. T., Demuth, H. B., Beale, M. H., and De Jesús, O. (1996). *Neural network design*, volume 20. PWS publishing company Boston.

Hagan, M. T., Demuth, H. B., and Jesús, O. D. (2002). An introduction to the use of neural networks in control systems. *International Journal of Robust and Nonlinear Control*, 12(11):959–985.

Hagan, M. T. and Menhaj, M. B. (1994). Training feedforward networks with the marquardt algorithm. *IEEE transactions on Neural Networks*, 5(6):989–993.

Haykin, S. and Li, X. B. (1995). Detection of signals in chaos. *Proceedings of the IEEE*, 83(1):95–122.

Haykin, S. and Principe, J. (1998). Making sense of a complex world [chaotic events modeling]. *Signal Processing Magazine, IEEE*, 15(3):66–81.

Heimes, F. O. (2008). Recurrent neural networks for remaining useful life estimation. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pages 1–6. IEEE.

Heinz P. Bloch, F. K. G. (2012). *Machinery failure analysis and troubleshooting: practical machinery management for process plants*. Butterworth-Heinemann.

Heng, A., Zhang, S., Tan, A. C., and Mathew, J. (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, 23(3):724–739.

Heng, A. S. Y. (2009). *Intelligent prognostics of machinery health utilising suspended condition monitoring data*. PhD thesis, Queensland University of Technology.

Hernández, M. A. and Stolfo, S. J. (1998). Real-world data is dirty: Data cleansing and the merge/purge problem. *Data mining and knowledge discovery*, 2(1):9–37.

Hess, A., Calvello, G., and Dabney, T. (2004). Phm a key enabler for the jsf autonomic logistics support concept. In *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, volume 6, pages 3543–3550. IEEE.

Hess, A., Calvello, G., and Frith, P. (2005). Challenges, issues, and lessons learned chasing the" big p". real predictive prognostics. part 1. In *Aerospace Conference, 2005 IEEE*, pages 3610–3619. IEEE.

Ho, Y.-C. and Pepyne, D. L. (2001). Simple explanation of the no free lunch theorem of optimization. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 5, pages 4409–4414. IEEE.

Hogg, R. V. and Craig, A. T. (1995). *Introduction to mathematical statistics.(5"" edition)*. Upper Saddle River, New Jersey: Prentice Hall.

Holland, S. M. (2008). Principal components analysis (pca). *Department of Geology, University of Georgia, Athens, GA*, pages 30602–2501.

Horner, R., El-Haram, M., and Munns, A. (1997). Building maintenance strategy: a new management approach. *Journal of Quality in Maintenance Engineering*, 3(4):273–280.

Hu, C., Youn, B. D., and Chung, J. (2012a). A multiscale framework with extended kalman filter for lithium-ion battery soc and capacity estimation. *Applied Energy*, 92:694–704.

Hu, C., Youn, B. D., Wang, P., and Yoon, J. T. (2012b). Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Reliability Engineering & System Safety*, 103:120–135.

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688.

IATA (2011). Airline maintenance costs executive commentary. *Last Checked, June 2014, from http://www.iata.org/workgroups/ Documents/M-CTF/AMC_ExecComment_FY09. pdf.*

ISO-13381 (2015). Iso 13381-1:2015 condition monitoring and diagnostics of machines – prognostics – part 1: General guidelines. Technical report, International Organization for Standardization.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.

Jamshidi, M. (2008). *Systems of systems engineering: principles and applications*. CRC press.

Jardine, A. K., Lin, D., and Banjevic, D. (2006a). A review on machinery diagnostics and prognostics implementing condition-based. *Mechanical systems and signal processing*, 20(7):1483–1510.

Jardine, A. K., Lin, D., and Banjevic, D. (2006b). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, 20(7):1483–1510.

Jardine, A. K. and Tsang, A. H. (2013). *Maintenance, replacement, and reliability: theory and applications*. CRC press.

Jardine, A. K. S. (1973). *Maintenance, replacement and reliability*. New York, Halsted Press.

Javed, K., Gouriveau, R., Zemouri, R., and Zerhouni, N. (2012). Features selection procedure for prognostics: An approach based on predictability. *IFAC Proceedings Volumes*, 45(20):25–30.

Javed, K., Gouriveau, R., and Zerhouni, N. (2013). Novel failure prognostics approach with dynamic thresholds for machine degradation. In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, pages 4404–4409. IEEE.

Jensen, F. V. (1996). *An introduction to Bayesian networks*, volume 210. UCL press London.

Jianzhong, S., Hongfu, Z., Haibin, Y., and Pecht, M. (2010). Study of ensemble learning-based fusion prognostics. In *Prognostics and Health Management Conference, 2010. PHM'10.*, pages 1–7. IEEE.

Jolliffe, I. T. and Stephenson, D. B. (2003). *Forecast verification: a practitioner's guide in atmospheric science*. John Wiley & Sons.

Jones, J. (2010). Maintenance. Lecture/Class Notes, Warwick Manufacturing Group.

Jones, J., Warrington, L., and Davis, N. (2001). The use of a discrete event simulation to model the achievement of maintenance free operating time

for aerospace systems. In *Reliability and Maintainability Symposium, 2001. Proceedings. Annual*, pages 170–175. IEEE.

Juesas, P., Ramasso, E., Drujont, S., and Placet, V. (2016). On partially supervised learning and inference in dynamic bayesian networks for prognostics with uncertain factual evidence: Illustration with markov switching models.

Julier, S. J. and Uhlmann, J. K. (1997). New extension of the kalman filter to nonlinear systems. In *AeroSense 97*, pages 182–193. International Society for Optics and Photonics.

Kacprzynski, G., Sarlashkar, A., Roemer, M., Hess, A., and Hardman, B. (2004). Predicting remaining life by fusing the physics of failure modeling with diagnostics. *JOM Journal of the Minerals, Metals and Materials Society*, 56(3):29–35.

Kan, M. S., Tan, A. C., and Mathew, J. (2015). A review on prognostic techniques for non-stationary and non-linear rotating systems. *Mechanical Systems and Signal Processing*, 62:1–20.

Kans, M. and Ingwald, A. (2008). Common database for cost-effective improvement of maintenance performance. *International journal of production economics*, 113(2):734–747.

Kazmierczak, K. (1983). Application of autoregressive prognostic techniques in diagnostics. In *Proceedings of the Vehicle Diagnostics Conference, Tuczno, Poland*.

Kelly, A. (1989). Maintenance and its management. In *Conference Communication*. conference communication.

Kim, N.-H., An, D., and Choi, J.-H. (2017). *Prognostics and Health Management of Engineering Systems*. Springer.

Kim, S.-H., Cohen, M. A., and Netessine, S. (2007). Performance contracting in after-sales service supply chains. *Management Science*, 53(12):1843–1858.

Knapp, G. M., Javadpour, R., and Wang, H.-P. (2000). An artmap neural network-based machine condition monitoring system. *Journal of Quality in Maintenance Engineering*, 6(2):86–105.

Knapp, G. M. and Wang, H.-P. (1992). Machine fault classification: a neural network approach. *International Journal of Production Research*, 30(4):811–823.

Koppen, M. (2004). No-free-lunch theorems and the diversity of algorithms. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 1, pages 235–241. IEEE.

Kozlowski, J. D., Watson, M. J., Byington, C. S., Garga, A. K., and Hay, T. A. (2001). Electrochemical cell diagnostics using online impedance measurement, state estimation and data fusion techniques. In *Intersociety Energy Conversion Engineering Conference*, volume 2, pages 981–986. SAE; 1999.

Kramer, S. and Tumer, I. Y. (2009). Towards statecharts based failure propagation analysis for designing embedded phm systems. In *The 2009 Prognostics and Health Management Conference, PHM*, volume 9.

Krenker, A., Bester, J., and Kos, A. (2011). Introduction to the artificial neural networks. *Artificial neural networks: methodological advances and biomedical applications. InTech, Rijeka. ISBN*, pages 978–953.

Kurzke, J. (2017). Gasturb 13 manual, design and off-design performance of gas turbines. Technical report, GasTurb GmbH.

Lacaille, J. and Bellas, A. (2014). Online normalization algorithm for engine turbofan monitoring. Technical report, Snecma Moissy-Cramayel France.

Lam, J., Sankararaman, S., and Stewart, B. (2014). Enhanced trajectory based similarity prediction with uncertainty quantification. In *Annual Conference of the Prognostics and Health Management Society 2014*.

Lawrence, S., Giles, C. L., and Tsoi, A. C. (1997). Lessons in neural network training: Overfitting may be harder than expected. In *AAAI/IAAI*, pages 540–545. Citeseer.

Leao, B. P., Yoneyama, T., Rocha, G. C., and Fitzgibbon, K. T. (2008). Prognostics performance metrics and their relation to requirements, design, verification and cost-benefit. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pages 1–8. IEEE.

Lebold, M. and Thurston, M. (2001). Open standards for condition-based maintenance and prognostic systems. In *Maintenance and Reliability Conference (MARCON)*, volume 200. May.

Lee, H. L. and Rosenblatt, M. J. (1987). Simultaneous determination of production cycle and inspection schedules in a production system. *Management science*, 33(9):1125–1136.

Lee, J., Ni, J., Djurdjanovic, D., Qiu, H., and Liao, H. (2006). Intelligent prognostics tools and e-maintenance. *Computers in industry*, 57(6):476–489.

Lee, J., Qiu, H., Yu, G., Lin, J., et al. (2007). Bearing data set. *IMS, University of Cincinnati, NASA Ames Prognostics Data Repository, Rexnord Technical Services*.

Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., and Siegel, D. (2014). Prognostics and health management design for rotary machinery systems—

reviews, methodology and applications. *Mechanical systems and signal processing*, 42(1):314–334.

Lewis, K. (2017). Technical note - tn 016: 2017 maintenance requirements analysis manual. Technical report, the Asset Standards Authority (ASA), Australia.

Li, C. J. and Choi, S. (2002). Spur gear root fatigue crack prognosis via crack diagnosis and fracture mechanics. In *Proceedings of the 56th Meeting of the Society of Mechanical Failures Prevention Technology*, pages 311–320.

Li, C. J. and Lee, H. (2005). Gear fatigue crack prognosis using embedded model, gear dynamic model and fracture mechanics. *Mechanical systems and signal processing*, 19(4):836–846.

Li, Y. (2002). Performance-analysis-based gas turbine diagnostics: A review. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 216(5):363–377.

Li, Y., Billington, S., Zhang, C., Kurfess, T., Danyluk, S., and Liang, S. (1999). Adaptive prognostics for rolling element bearing condition. *Mechanical systems and signal processing*, 13(1):103–113.

Li, Y., Kurfess, T., and Liang, S. (2000). Stochastic prognostics for rolling element bearings. *Mechanical Systems and Signal Processing*, 14(5):747–762.

Li, Y. and Nilkitsaranont, P. (2009). Gas turbine performance prognostic for condition-based maintenance. *Applied energy*, 86(10):2152–2161.

Liao, H. and Sun, J. (2011). Nonparametric and semi-parametric sensor recovery in multichannel condition monitoring systems. *IEEE Transactions on Automation Science and Engineering*, 8(4):744–753.

Liao, L. and Köttig, F. (2014). Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Transactions on Reliability*, 63(1):191–207.

Lin, Y., Chen, M., and Zhou, D. (2013). Online probabilistic operational safety assessment of multi-mode engineering systems using bayesian methods. *Reliability Engineering & System Safety*, 119:150–157.

Lippmann, R. P. (1988). An introduction to computing with neural nets. *ACM SIGARCH Computer Architecture News*, 16(1):7–25.

Liu, C., Ihiabe, D., Laskaridis, P., and Singh, R. (2013a). A preliminary method to estimate impacts of inlet flow distortion on boundary layer ingesting propulsion system design point performance. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, page 0954410013496750.

Liu, K., Gebraeel, N. Z., and Shi, J. (2013b). A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. *IEEE Transactions on Automation Science and Engineering*, 10(3):652–664.

Liu, Z., Yin, X., Zhang, Z., Chen, D., and Chen, W. (2004). Online rotor mixed fault diagnosis way based on spectrum analysis of instantaneous power in squirrel cage induction motors. *IEEE Transactions on Energy Conversion*, 19(3):485–490.

Loutas, T., Eleftheroglou, N., and Zarouchas, D. (2017). A data-driven probabilistic framework towards the in-situ prognostics of fatigue life of composites based on acoustic emission data. *Composite Structures*, 161:522–529.

Luchinsky, D., Osipov, S., Smelyanskiy, V., Kiris, C., Timucin, D., and Lee, S. H. (2007). In-flight failure decision and prognostic for the solid rocket

buster. In *43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, page 5823.

MacKay, D. J. (1992a). Bayesian interpolation. *Neural computation*, 4(3):415–447.

MacKay, D. J. (1992b). A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472.

Majidian, A. and Saidi, M. (2007). Comparison of fuzzy logic and neural network in life prediction of boiler tubes. *International Journal of Fatigue*, 29(3):489–498.

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., and Winkler, R. (1982). The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of forecasting*, 1(2):111–153.

Malinowski, S., Chebel-Morello, B., and Zerhouni, N. (2015). Remaining useful life estimation based on discriminating shapelet extraction. *Reliability Engineering & System Safety*, 142:279–288.

Marble, S. and Morton, B. P. (2006). Predicting the remaining life of propulsion system bearings. In *Aerospace Conference, 2006 IEEE*, pages 8–pp. IEEE.

Marinai, L., Probert, D., and Singh, R. (2004). Prospects for aero gas-turbine diagnostics: a review. *Applied Energy*, 79(1):109–126.

MathWorks, I. (2002). *Curve fitting toolbox: for use with MATLAB®: user's guide*. MathWorks.

Mazdiyasni, S. (2008). Opportunities and challenges in nde and health monitoring of turbine engine components. In *AIP Conference Proceedings*, volume 975, pages 1468–1475. AIP.

McLachlan, G. and Peel, D. (2004). *Finite mixture models*. John Wiley & Sons.

Medjaher, K., Zerhouni, N., and Baklouti, J. (2013). Data-driven prognostics based on health indicator construction: Application to pronostia's data. In *Control Conference (ECC), 2013 European*, pages 1451–1456. IEEE.

Meier, G. and Pettit, F. (1989). High-temperature corrosion of alumina-forming coatings for superalloys. *Surface and Coatings Technology*, 39:1–17.

Menezes, J. M. P. and Barreto, G. A. (2008). Long-term time series prediction with the narx network: an empirical evaluation. *Neurocomputing*, 71(16):3335–3343.

MentorGraphics (2013). Flowmaster, gas turbine system modeling & simulation. *Last checked, November 2013, from http://www.mentor.com/products/mechanical/flowmaster/flowmaster-process-power-energy/*.

Miao, Q., Xie, L., Cui, H., Liang, W., and Pecht, M. (2013). Remaining useful life prediction of lithium-ion battery with unscented particle filter technique. *Microelectronics Reliability*, 53(6):805–810.

Milano, D., Scannapieco, M., and Catarci, T. (2005). Using ontologies for xml data cleaning. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 562–571. Springer.

Misra, K. B. (2008). Maintenance engineering and maintainability: An introduction. In *Handbook of performability engineering*, pages 755–772. Springer.

Morando, S., Jemei, S., Gouriveau, R., Zerhouni, N., and Hissel, D. (2013). Fuel cells prognostics using echo state network. In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, pages 1632–1637. IEEE.

Mosallam, A., Medjaher, K., and Zerhouni, N. (2016). Data-driven prognostic method based on bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing*, 27(5):1037–1048.

Moubray, J. (1997). *Reliability-centered maintenance*. Industrial Press Inc.

Moyle, S. (2009). Collaborative data mining. In *Data Mining and Knowledge Discovery Handbook*, pages 1029–1039. Springer.

Muller, A., Suhner, M.-C., and Iung, B. (2008). Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system. *Reliability Engineering & System Safety*, 93(2):234–253.

Murata, N., Yoshizawa, S., and Amari, S.-i. (1994). Network information criterion-determining the number of hidden units for an artificial neural network model. *Neural Networks, IEEE Transactions on*, 5(6):865–872.

Nakagawa, T. (1981). A summary of periodic replacement with minimal repair at failure. *Journal of the Operations Research Society of Japan*, 24(3):213–227.

Nakagawa, T. and Yasui, K. (1991). Periodic-replacement models with threshold levels. *IEEE Transactions on Reliability*, 40(3):395–397.

Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Morello, B., Zerhouni, N., and Varnier, C. (2012). Pronostia: An experimental platform for bearing accelerated degradation tests. In *IEEE International Conference on Prognostics and Health Management, Denver, CO, USA*.

Oppenheimer, C. H. and Loparo, K. A. (2002). Physically based diagnosis and prognosis of cracked rotor shafts. In *AeroSense 2002*, pages 122–132. International Society for Optics and Photonics.

Orchard, M., Wu, B., and Vachtsevanos, G. (2005). A particle filtering framework for failure prognosis. In *World Tribology Congress III*, pages 883–884. American Society of Mechanical Engineers.

Orchard, M. E. and Vachtsevanos, G. J. (2009). A particle-filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*.

Orhan, E. (2012). Particle filtering. *Center for Neural Science, University of Rochester, Rochester, NY*, 8(11).

Orsagh, R., Roemer, M., Sheldon, J., and Klenke, C. J. (2004). A comprehensive prognostics approach for predicting gas turbine engine bearing life. In *ASME Turbo Expo 2004: Power for Land, Sea, and Air*, pages 777–785. American Society of Mechanical Engineers.

Orsagh, R. F., Sheldon, J., and Klenke, C. J. (2003). Prognostics/diagnostics for gas turbine engine bearings. In *ASME Turbo Expo 2003, collocated with the 2003 International Joint Power Generation Conference*, pages 159–167. American Society of Mechanical Engineers.

Paris, P. C. and Erdogan, F. (1963). A critical analysis of crack propagation laws. ASME.

Parker Jr, B. E., Nigro, T. M., Carley, M. P., Barron, R. L., Ward, D. G., Poor, H. V., Rock, D., and DuBois, T. A. (1993). Helicopter gearbox diagnostics and prognostics using vibration signature analysis. In *Optical Engineering and Photonics in Aerospace Sensing*, pages 531–542. International Society for Optics and Photonics.

Pastor-Fernández, C., Widanage, W. D., Chouchelamane, G., and Marco, J. (2016). A soh diagnosis and prognosis method to identify and quantify degradation modes in li-ion batteries using the ic/dv technique. In *IET Hybrid Electric Vehicle Conference*. IET.

Patel, V., Kadirkamanathan, V., and Thompson, H. (1996). A novel self-learning fault detection system for gas turbine engines.

PCoE (2008). Pcoe datasets. *NASA Ames Prognostics Data Repository (https://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/), NASA Ames Research Center, Moffett Field, CA*.

PCoE (2014). Pcoe datasets. *NASA Ames Prognostics Data Repository (https://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/), NASA Ames Research Center, Moffett Field, CA*.

Pecht, M. (2008). *Prognostics and health management of electronics*. Wiley Online Library.

Peel, L. (2008). Data driven prognostics using a kalman filter ensemble of neural network models. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pages 1–6. IEEE.

Peng, Y., Dong, M., and Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, 50(1-4):297–313.

Peng, Y., Wang, H., Wang, J., Liu, D., and Peng, X. (2012a). A modified echo state network based remaining useful life estimation approach. In *Prognostics and Health Management (PHM), 2012 IEEE Conference on*, pages 1–7. IEEE.

Peng, Y., Xu, Y., Liu, D., and Peng, X. (2012b). Sensor selection with grey correlation analysis for remaining useful life evaluation. In *Annual conference of the phm society*.

Penna, J. A. M., Nascimento, C. L., and Rodrigues, L. R. (2012). Health monitoring and remaining useful life estimation of lithium-ion aeronautical batteries. In *Aerospace Conference, 2012 IEEE*, pages 1–12. IEEE.

PlantServices (2004). The basics of predictive/preventive maintenance. *Last checked = 24 October 2013, from http://www.plantservices.com/ whitepapers/2004/28/*.

Principe, J. C., Euliano, N. R., and Lefebvre, W. C. (1999). *Neural and adaptive systems: fundamentals through simulations with CD-ROM*. John Wiley & Sons, Inc.

Pusey, H. C. (1999). An assessment of turbomachinery condition monitoring and failure prognosis technology. In *SPIE proceedings series*, pages 1310–1316. Society of Photo-Optical Instrumentation Engineers.

Raheja, D., Llinas, J., Nagi, R., and Romanowski, C. (2000). A holistic view of condition based maintenance. Technical report, Working Paper, December 2000 (Center for Multisource Information Fusion (CMIF), SUNY.

Raheja, D., Llinas, J., Nagi, R., and Romanowski, C. (2006). Data fusion/data mining-based architecture for condition-based maintenance. *International Journal of Production Research*, 44(14):2869–2887.

Ramasso, E. (2009). Contribution of belief functions to hidden markov models with an application to fault diagnosis. In *Machine Learning for Signal Processing, 2009. MLSP 2009. IEEE International Workshop on*, pages 1–6. IEEE.

Ramasso, E. (2014a). Investigating computational geometry for failure prognostics. *International Journal of Prognostics and Health Management*, 5(1):005.

Ramasso, E. (2014b). Investigating computational geometry for failure prognostics in presence of imprecise health indicator: Results and comparisons on c-mapss datasets. *European conf. on prognostics and health management.*, 5(1):005.

Ramasso, E. and Gouriveau, R. (2010). Prognostics in switching systems: Evidential markovian classification of real-time neuro-fuzzy predictions. In *Prognostics and Health Management Conference, 2010. PHM'10.*, pages 1–10. IEEE.

Ramasso, E. and Saxena, A. (2014). Performance benchmarking and analysis of prognostic methods for cmapss datasets. *International Journal of Prognostics and Health Management*, 5(2):1–15.

Rareshide, E., Tindal, A., Johnson, C., Graves, A., Simpson, E., Bleeg, J., Harris, T., and Schoborg, D. (2009). Effects of complex wind regimes on turbine performance. In *Proceedings American Wind Energy Association WINDPOWER Conference (Chicago, IL)*.

Rezvani, M., Lee, S., and Lee, J. (2011). A comparative analysis of techniques for electric vehicle battery prognostics and health management (phm). Technical report, SAE Technical Paper.

Riad, A., Elminir, H., and Elattar, H. (2010). Evaluation of neural networks in the subject of prognostics as compared to linear regression model. *International Journal of Engineering & Technology*, 10(1):52–58.

Rigamonti, M., Baraldi, P., Zio, E., Roychoudhury, I., Goebel, K., and Poll, S. (2016). Echo state network for the remaining useful life prediction of

a turbofan engine. In *Proceedings of the Third European Prognostic and Health Management Conference, PHME*.

Rigamonti, M., Baraldi, P., Zio, E., Roychoudhury, I., Goebel, K., and Poll, S. (2017). Ensemble of optimized echo state networks for remaining useful life prediction. *Neurocomputing*.

Roemer, M. J., Byington, C. S., Kacprzynski, G. J., and Vachtsevanos, G. (2006). An overview of selected prognostic technologies with application to engine health management. In *ASME Turbo Expo 2006: Power for Land, Sea, and Air*, pages 707–715. American Society of Mechanical Engineers.

Rubini, R. and Meneghetti, U. (2001). Application of the envelope and wavelet transform analyses for the diagnosis of incipient faults in ball bearings. *Mechanical systems and signal processing*, 15(2):287–302.

Saha, B. and Goebel, K. (2007). Battery data set. *NASA Ames Prognostics Data Repository (http://ti. arc. nasa. gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA*.

Saha, B. and Goebel, K. (2009). Modeling li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the annual conference of the prognostics and health management society*, pages 2909–2924.

Saha, B., Goebel, K., Poll, S., and Christophersen, J. (2009). Prognostics methods for battery health monitoring using a bayesian framework. *IEEE Transactions on instrumentation and measurement*, 58(2):291–296.

Sally, J. D. (2007). *Roots to research: a vertical development of mathematical problems*, chapter 3, page 63. American Mathematical Soc.

Sandborn, P. A. and Wilkinson, C. (2007). A maintenance planning and business case development model for the application of prognostics and

health management (phm) to electronic systems. *Microelectronics Reliability*, 47(12):1889–1901.

Sankararaman, S. and Goebel, K. (2015). Uncertainty in prognostics and systems health management. *Int J Prognostic Health Manag*, 6(010).

Sarkar, S., Jin, X., and Ray, A. (2011). Data-driven fault detection in aircraft engines with noisy sensor measurements. *Journal of Engineering for Gas Turbines and Power*, 133(8):081602.

Sarle, W. S. (1994). Neural networks and statistical models. *Proceedings of the Nineteenth Annual SAS Users Group International Conference*.

Satish, B. and Sarma, N. (2005). A fuzzy bp approach for diagnosis and prognosis of bearing faults in induction motors. In *Power Engineering Society General Meeting, 2005. IEEE*, pages 2291–2294. IEEE.

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., and Schwabacher, M. (2008a). Metrics for evaluating performance of prognostic techniques. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pages 1–17. IEEE.

Saxena, A., Celaya, J., Saha, B., Saha, S., and Goebel, K. (2009a). Evaluating algorithm performance metrics tailored for prognostics. In *Aerospace conference, 2009 IEEE*, pages 1–13. IEEE.

Saxena, A., Celaya, J., Saha, B., Saha, S., and Goebel, K. (2009b). On applying the prognostic performance metrics. *Annual Conference of the Prognostics and Health Management Society, 2009*.

Saxena, A. and Goebel, K. (2008a). Phm08 challenge data set. *NASA Ames Prognostics Data Repository (http://ti. arc. nasa. gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA*.

Saxena, A. and Goebel, K. (2008b). Turbofan engine degradation simulation data set. *NASA Ames Prognostics Data Repository (http://ti. arc. nasa. gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA*.

Saxena, A., Goebel, K., Simon, D., and Eklund, N. (2008b). Damage propagation modeling for aircraft engine run-to-failure simulation. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pages 1–9. IEEE.

Saxena, A., Sankararaman, S., and Goebel, K. (2014). Performance evaluation for fleet-based and unit-based prognostic methods. In *Second European conference of the Prognostics and Health Management society*, pages 8–10.

Schoen, R. R. and Habetler, T. G. (1993). Effects of time-varying loads on rotor fault detection in induction machines. In *Industry Applications Society Annual Meeting, 1993., Conference Record of the 1993 IEEE*, pages 324–330. IEEE.

Schwabacher, M. (2005). A survey of data-driven prognostics. In *Infotech@ Aerospace*, page 7002.

Schwabacher, M. and Goebel, K. (2007). A survey of artificial intelligence for prognostics. In *Aaai fall symposium*, pages 107–114.

Seo, J. (2005). *Information visualization design for multidimensional data: integrating the rank-by-feature framework with hierarchical clustering*. PhD thesis.

Sheppard, J. W., Kaufman, M. A., and Wilmering, T. J. (2008). Ieee standards for prognostics and health management. In *AUTOTESTCON, 2008 IEEE*, pages 97–103. IEEE.

Sheut, C. and Krajewski, L. (1994). A decision model for corrective maintenance management. *The International Journal of Production Research*, 32(6):1365–1382.

Shin, J.-H. and Jun, H.-B. (2015). On condition based maintenance policy. *Journal of Computational Design and Engineering*, 2(2):119–127.

Si, X.-S., Wang, W., Hu, C.-H., and Zhou, D.-H. (2011). Remaining useful life estimation–a review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1):1–14.

Siegel, D. (2009). *Evaluation of health assessment techniques for rotating machinery*. PhD thesis, University of Cincinnati.

Sikorska, J., Hodkiewicz, M., and Ma, L. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, 25(5):1803–1836.

Simões, J. M., Gomes, C. F., and Yasin, M. M. (2011). A literature review of maintenance performance measurement: A conceptual framework and directions for future research. *Journal of Quality in Maintenance Engineering*, 17(2):116–137.

Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., and Lendasse, A. (2007). Methodology for long-term prediction of time series. *Neurocomputing*, 70(16):2861–2869.

Soukhanov, A. H. (2001). *Microsoft Encarta College Dictionary: The First Dictionary for the Internet Age*. Macmillan.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Staszewski, W. and Tomlinson, G. (1994). Application of the wavelet transform to fault detection in a spur gear. *Mechanical Systems and Signal Processing*, 8(3):289–307.

Sugeno, M., Griffin, M., and Bastian, A. (1993). Fuzzy hierarchical control of an unmanned helicopter. In *17th IFSA World Congress*, pages 179–182.

Sun, B., Zeng, S., Kang, R., and Pecht, M. (2010). Benefits analysis of prognostics in systems. In *Prognostics and Health Management Conference, 2010. PHM'10.*, pages 1–8. IEEE.

Swanson, D. C. (2001). A general prognostic tracking algorithm for predictive maintenance. In *Aerospace Conference, 2001, IEEE Proceedings.*, volume 6, pages 2971–2977. IEEE.

Tamilselvan, P. and Wang, P. (2013). Failure diagnosis using deep belief learning based health state classification. *Reliability Engineering & System Safety*, 115:124–135.

Tobon-Mejia, D. A., Medjaher, K., and Zerhouni, N. (2010). The iso 13381-1 standard's failure prognostics process through an example. In *Prognostics and Health Management Conference, 2010. PHM'10.*, pages 1–12. IEEE.

Tobon-Mejia, D. A., Medjaher, K., Zerhouni, N., and Tripot, G. (2012). A data-driven failure prognostics method based on mixture of gaussians hidden markov models. *IEEE Transactions on Reliability*, 61(2):491–503.

Tran, V. T. and Yang, B.-S. (2012). An intelligent condition-based maintenance platform for rotating machinery. *Expert Systems with Applications*, 39(3):2977–2988.

Tsang, A. H. (1995). Condition-based maintenance: tools and decision making. *Journal of Quality in Maintenance Engineering*, 1(3):3–17.

Tsang, A. H. (2002). Strategic dimensions of maintenance management. *Journal of Quality in Maintenance Engineering*, 8(1):7–39.

Tsang, A. H., Jardine, A. K., and Kolodny, H. (1999). Measuring maintenance performance: a holistic approach. *International Journal of Operations & Production Management*, 19(7):691–715.

Tumer, I. Y. and Bajwa, A. (1999). A survey of aircraft engine health monitoring systems. In *Proc. 35th Joint Propulsion Conf.*

Tumer, I. Y. and Huff, E. M. (2003). Analysis of triaxial vibration data for health monitoring of helicopter gearboxes. *Journal of vibration and acoustics*, 125(1):120–128.

Uckun, S., Goebel, K., and Lucas, P. J. (2008). Standardizing research methods for prognostics. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pages 1–10. IEEE.

Ultsch, A. (1993). Self-organizing neural networks for visualisation and classification. In *Information and classification*, pages 307–313. Springer.

Uluyol, O., Parthasarathy, G., Foslien, W., and Kim, K. (2011). Power curve analytic for wind turbine performance monitoring and prognostics. In *Annual conference of the prognostics and health management society*, volume 2, pages 1–8.

Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., and Wu, B. Intelligent fault diagnosis and prognosis for engineering systems, 2006. *Usa 454p Isbn*, 13:978–0.

Vassiliadis, P. (1998). Modeling multidimensional databases, cubes and cube operations. In *Scientific and Statistical Database Management, 1998. Proceedings. Tenth International Conference on*, pages 53–62. IEEE.

Vatani, A. (2013). *Degradation Prognostics in Gas Turbine Engines Using Neural Networks*. PhD thesis, Concordia University.

Venkatasubramanian, V. (2005). Prognostic and diagnostic monitoring of complex systems for product lifecycle management: Challenges and opportunities. *Computers & chemical engineering*, 29(6):1253–1263.

Virkler, D., Hillberry, B., and Goel, P. (1979). The statistical nature of fatigue crack propagation. *Trans. ASME, J. Engng Mater. Technol. 101, 148-153.*

Visser, W. P. and Broomhead, M. J. (2000). Gsp, a generic object-oriented gas turbine simulation environment. In *ASME Turbo Expo 2000: Power for Land, Sea, and Air*, pages V001T01A002–V001T01A002. American Society of Mechanical Engineers.

Volponi, A. (2005). Data fusion for enhanced aircraft engine prognostics and health management. *Glenn Research Center, Cleveland, OH, NASA Report No. CR-2005-214055.*

Volponi, A. J. (2014). Gas turbine engine health management: Past, present, and future trends. *Journal of Engineering for Gas Turbines and Power*, 136(5):051201.

Wang, P. and Gao, R. X. (2014). Particle filtering-based system degradation prediction applied to jet engines. In *Annual Conference of the Prognostics and Health Management Society*, page 1. Citeseer.

Wang, P., Youn, B. D., and Hu, C. (2012). A generic probabilistic framework for structural health prognostics and uncertainty management. *Mechanical Systems and Signal Processing*, 28:622–637.

Wang, T. (2010). *Trajectory similarity based prediction for remaining useful life estimation*. PhD thesis, University of Cincinnati.

Wang, T., Yu, J., Siegel, D., and Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pages 1–6. IEEE.

Wang, W. and McFadden, P. (1996). Application of wavelets to gearbox vibration signals for fault detection. *Journal of sound and vibration*, 192(5):927–939.

Wang, W. Q., Golnaraghi, M. F., and Ismail, F. (2004). Prognosis of machine health condition using neuro-fuzzy systems. *Mechanical Systems and Signal Processing*, 18(4):813–831.

Wei, W. W.-S. (1994). *Time series analysis*. Addison-Wesley publ Reading.

Whitle, P. (1951). *Hypothesis testing in time series analysis*, volume 4. Almqvist & Wiksells.

Wood, B. (1999). Intelligent building care. *Facilities*, 17(5/6):189–194.

Wu, J., Zhang, C., and Chen, Z. (2016). An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks. *Applied energy*, 173:134–140.

Wu, W., Hu, J., and Zhang, J. (2007). Prognostics of machine health condition using an improved arima-based prediction method. In *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*, pages 1062–1067. Ieee.

Xue, Y., Williams, D. P., and Qiu, H. (2011). Classification with imperfect labels for fault prediction. In *Proceedings of the first international workshop on data mining for service and maintenance*, pages 12–16. ACM.

Yan, J., Koc, M., and Lee, J. (2004). A prognostic algorithm for machine performance assessment and its application. *Production Planning & Control*, 15(8):796–801.

Yang, F., Habibullah, M. S., Zhang, T., Xu, Z., Lim, P., and Nadarajan, S. (2016). Health index-based prognostics for remaining useful life predictions in electrical machines. *IEEE Transactions on Industrial Electronics*, 63(4):2633–2644.

Yu, J. (2013). A nonlinear probabilistic method and contribution analysis for machine condition monitoring. *Mechanical Systems and Signal Processing*, 37(1):293–314.

Zaidan, M. A. (2014). *Bayesian Approaches for Complex System Prognostics*. PhD thesis, University of Sheffield.

Zaidan, M. A., Mills, A. R., and Harrison, R. F. (2013). Bayesian framework for aerospace gas turbine engine prognostics. In *Aerospace Conference, 2013 IEEE*, pages 1–8. IEEE.

Zemouri, R., Gouriveau, R., and Zerhouni, N. (2010). Defining and applying prediction performance metrics on a recurrent narx time series model. *Neurocomputing*, 73(13):2506–2521.

Zhang, S., Ma, L., Sun, Y., and Mathew, J. (2007). Asset health reliability estimation based on condition data. *Proceedings World Congress on Engineering Asset Management,*.

Zhao, D., Georgescu, R., and Willett, P. (2011). Comparison of data reduction techniques based on svm classifier and svr performance. In *SPIE Optical Engineering+ Applications*, pages 81370X–81370X. International Society for Optics and Photonics.

Zheng, S., Ristovski, K., Farahat, A., and Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. In *Prognostics and Health Management (ICPHM), 2017 IEEE International Conference on*, pages 88–95. IEEE.

Appendix A

A.1 Data Challenge Leader Board and Performance Metrics

Pub. Ref	Metric	FD001	FD002	FD003	FD004	PHM08 test	PHM08 final test
Ramasso (2014a)	PHM08 Score	216	2796	317	3132	752	11572
	FPR& FNR	56-44	51-49	66-34	49-51	-	-
	MAE	10	17	12	18	-	-
	MSE	176	524	256	592	-	-
Wang et al. (2012)	PHM08 Score	-	-	-	-	1139	-
Hu et al. (2012b)	PHM08 Score	-	-	-	-	1134	-
Wang et al. (2008)	PHM08 Score	-	-	-	-	-	5636.06
Heimes (2008)	PHM08 Score	-	-	-	-	519.8	6691.86
Peel (2008)	PHM08 Score	-	-	-	-	-	8637.57
Riad et al. (2010)	PHM08 Score	-	-	-	-	1540	-
Peng et al. (2012a)	MSE	3969	-	-	-	-	-
Liu et al. (2013b)	MAPE	9	-	-	-	-	-
Coble and Hines (2011)	PHM08 Score	-	-	-	-	2500	-

Table A.1: Performance of selected approaches

Rank on Final Test	Score	MSE	FPR (%)	FNR (%)	MAPE (%)	MAE	Corr. Score	Std. Dev.	MAD	MdAD
1	5636.06	546.60	64.83	31.72	19.99	16.23	0.93	1.01	16.33	11.00
2	6691.86	560.12	63.68	36.32	17.92	15.38	0.94	1.03	16.29	8.08
3	8637.57	672.17	61.38	23.45	20.72	17.69	0.92	1.09	17.79	11.00
4	9530.35	741.20	58.39	39.54	34.93	20.19	0.90	1.22	20.17	15.00
5	10571.58	764.82	58.85	41.15	32.60	20.05	0.91	1.22	20.41	14.23
6	14275.60	716.65	59.77	37.01	21.61	18.16	0.90	1.17	18.57	11.00
7	19148.88	822.06	56.09	41.84	30.25	20.23	0.88	1.29	20.89	13.00
8	20471.33	1000.06	51.95	48.05	33.63	22.44	0.88	1.42	24.05	14.78
9	22755.85	1078.19	62.53	35.40	39.90	24.51	0.86	1.45	24.08	20.00
10	25921.26	854.57	34.25	64.83	51.38	22.66	0.86	1.36	21.49	16.00

Rank on Test	Score	MSE	FPR (%)	FNR (%)	MAPE (%)	MAE	Corr. Score	Std. Dev.	MAD	MdAD
1	512.12	152.71	56.35	43.65	15.81	8.67	0.96	0.64	8.68	5.69
2	740.31	224.79	57.73	38.12	18.92	10.77	0.94	0.76	10.72	7.00
3	873.36	265.62	53.59	28.18	19.19	11.47	0.93	0.81	11.75	8.00
4	1218.43	269.68	51.93	44.20	20.15	11.87	0.93	0.85	12.05	8.00
5	1218.76	331.30	50.55	49.45	33.14	13.81	0.91	0.95	14.03	10.87
6	1232.27	334.52	42.27	57.73	32.90	14.14	0.91	0.96	14.28	10.37
7	1568.98	394.46	50.55	47.24	36.75	15.37	0.89	1.03	15.48	12.00
8	1645.77	330.02	47.24	50.28	30.00	13.47	0.91	0.95	13.59	10.00
9	1816.60	359.97	50.28	49.17	26.47	13.82	0.90	0.99	14.07	9.75
10	1839.06	377.01	45.86	53.59	27.72	14.31	0.89	1.02	14.43	9.10

Table A.2: PHM08 Data Challenge Leader board

A.2 Multi Regime Data

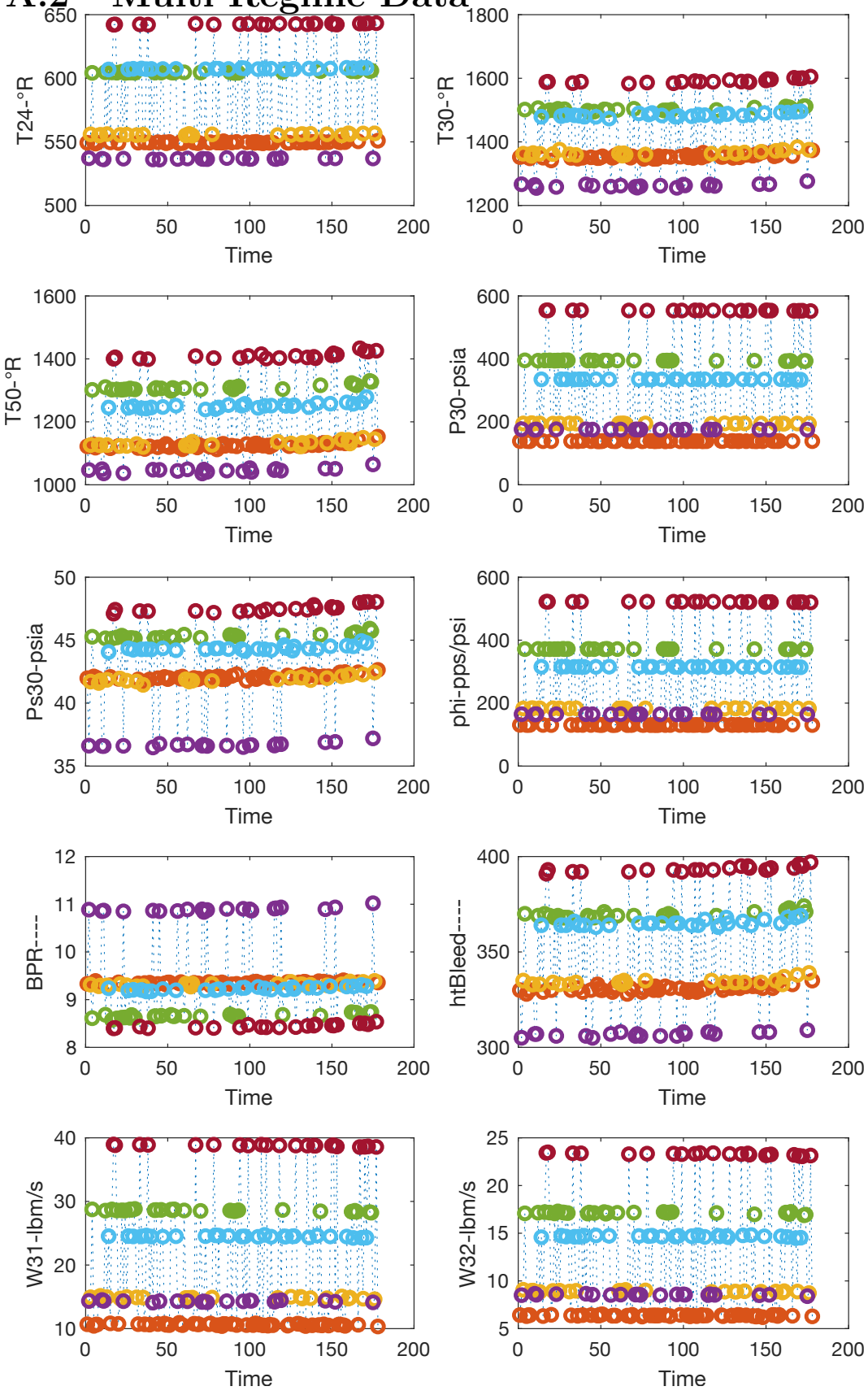

Figure A.1: Clustered Regimes for Raw Sensors

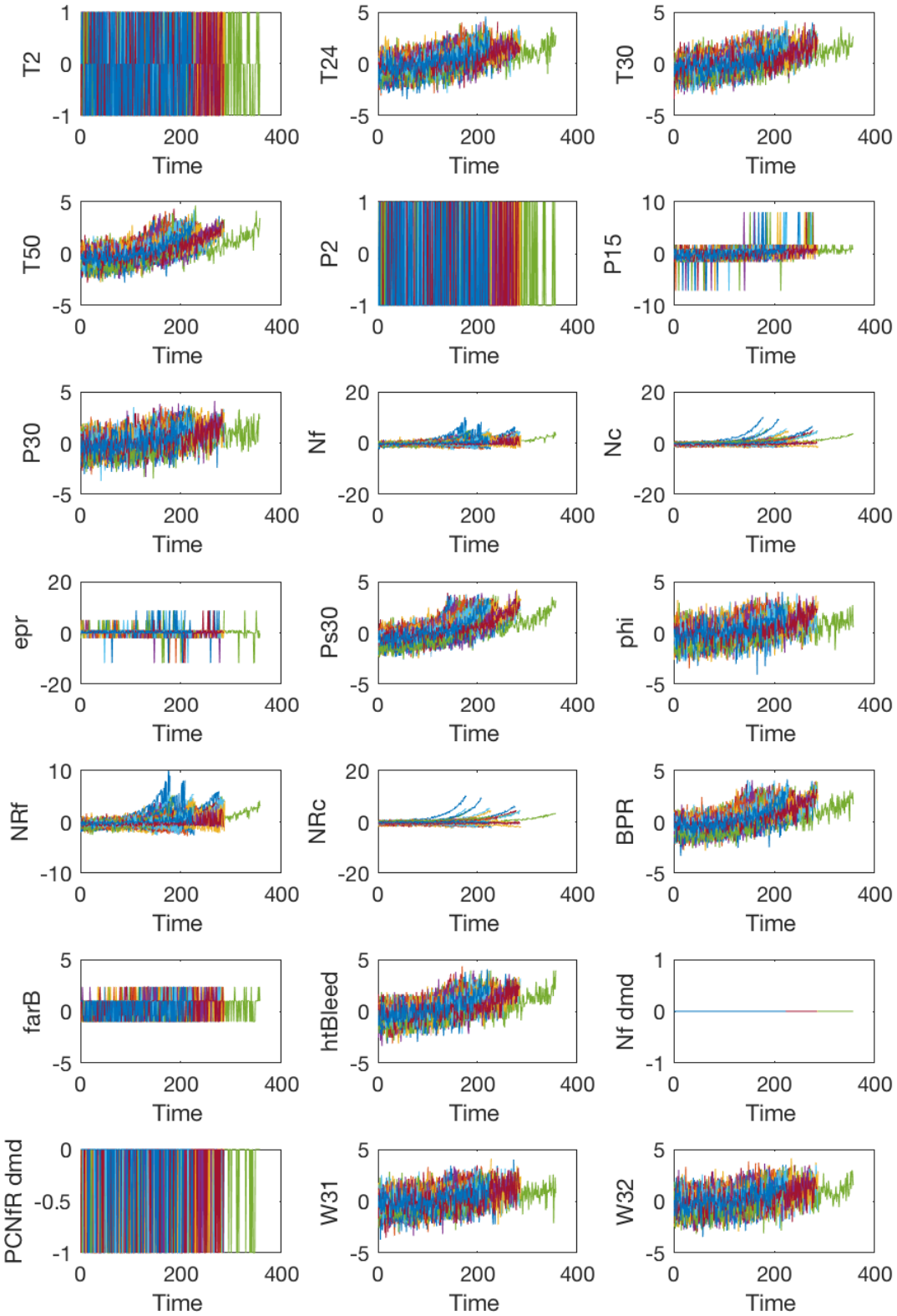

Figure A.2: Sensor behaviours in different trajectories

Appendix B

B.1 MATLAB Code for Synthetic Data Generation

```
function [data_proc,y,yp]=SyntheticDataGen(data_raw,ref,eq,
    eq2)
%% SyntheticDataGen >>Synthetic Data Generation
%  This function generates synthetic data
%% data_raw >> baseline data
% ref >> the sensor with constant values
%[data_proc_b,yb,ypb]=SyntheticDataGen(data_raw,ref,'f=f/log
    (0.95);','y=exp(-y/f);'); calculates an exponential
    baseline curve
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Oguz BEKTAS
% The University of Warwick
% The Warwick Manufacturing Group
% Coventry CV4 7AL, UK
% Last Update: 07/06/2016
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    unref=unique(ref);
    %% Synthetic Baseline
    y=[size(data_raw,1):-1:1]'-1;
    f=95*length(y)/100; f=f/max(y);
    % evaluate the first equation >> 'f=f/log(0.05)'
    eval(eq);
    y=y/max(y);
    % evaluate the second equation >> 'y=exp(-y/f);'
    eval(eq2);

    y=y/max(y);

```

```
for i=1:size(data_raw,2)
    yt=y;
    for j=1:length(unref)
    % regime clustering
    cluster=find(ref==(unref(j)));
    % 4th degree polynomial is used to fit the data
    expdata=Fun_fit(data_raw(cluster,i),'poly4');
    % identify the noise to add back later
    Noise=expdata-data_raw(cluster,i);
        % Normalise to [0, 1]:
    m = min(yt);
    range = max(yt) - m;
    yt = (yt - m) / range;
    % Then scale to [x,y]:
    range2 = max(expdata) - min(expdata);
    %% Noise added
    data_proc(cluster,i) = (yt(cluster)*range2) + min(
       expdata)+Noise;
    clear cluster expdata m range range2 Noise
    end
end
end


function expo_fit_v=Fun_fit(array,expt)
expo_fit_v=array;
expo_fit_v_m=expo_fit_v;
sza2=size(array);
Yeksn=(1:length(array))';
for i=1:sza2(2)
f_1_f = fit(Yeksn,array(1:end,i),expt);

expo_fit_v(1:end,i) = feval(f_1_f,Yeksn);

end
end
```

Appendix C

C.1 MATLAB Code for Secondary Data Adaptation

```
clear all; close all; clc;
%% Secondary Data Generation
% Oguz BEKTAS
% The University of Warwick
% The Warwick Manufacturing Group
% Coventry CV4 7AL, UK
% Last Update: 07/06/2016
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Data is imported from the text file
% train_data includes all "training" trajectories
% test_data includes all "final test" trajectories
% The first column stands for the unit number

%% Import data from text file.
disp('Please, choose PHM08 dataset directory')
DirectoryOfDataset= uigetdir('...','Choose directory
    containing CMAPSS data');

train_data=dlmread([DirectoryOfDataset '/train.txt'],' ');

test_data=dlmread([DirectoryOfDataset '/final_test.txt'],' '
    );

% Assign LongTrim,ShortTrim,ShortTrim2
produced_data=SecondaryDataGen(train_data,test_data,LongTrim
    ,ShortTrim,ShortTrim2)

%% function SecondaryDataGen
```

```
function produced_data=SecondaryDataGen(train_data,test_data
    ,LongTrim,ShortTrim,ShortTrim2)

for i=1:size(unique(test_data(:,1)))
    % clustering the training trajectories
    tr=find(train_data(:,1)==i);
    % clustering the test trajectories
    te=find(test_data(:,1)==i);
try % The train data has less trajectories
    % The first column is the raw training data
    produced_data{i,1}=train_data(tr,:);
    produced_datasize(i,1)=size(produced_data{i,1},1);
end
    % The second column is the raw test data
    produced_data{i,2}=test_data(te,:);
    produced_datasize(i,2)=size( produced_data{i,2},1);
      clear tr te simple1 simple x x1
end
 for i=1:size(produced_data,1)
    clear Kn BiggerTrain BaseTrainsize L1 StartBaseTrainsize
    % Length of original test trajectory
    Kn=size(produced_data{i, 2},1);
    % The long test trajectories are trimmed
if Kn>LongTrim Kn=LongTrim; end
    % Find training samples longer than the test
    BiggerTrain=find(produced_datasize(:,1)>Kn+5);
    % and random selection
    BaseTrain=randsample(BiggerTrain,1);
    % Modify the unit number (first column)
    BaseTrainsize(1:size(produced_data{BaseTrain, 1},1 ),1)
        =1:size(produced_data{BaseTrain, 1},1 );
% In the original data,short samples was not fond in high
    wear levels
if Kn<ShortTrim
    L1=round((size(BaseTrainsize,1)-Kn)*(1/2));
    % random location
    StartBaseTrainsize=randsample(BaseTrainsize(1:L1),1);
    elseif  Kn>=ShortTrim && Kn<ShortTrim2
    L1=round((size(BaseTrainsize,1)-Kn)*(2/3));
    % random location
    StartBaseTrainsize=randsample(BaseTrainsize(1:L1),1);
    else
    L1=((size(BaseTrainsize,1)-Kn));
    % random location
    StartBaseTrainsize=randsample(BaseTrainsize(1:L1-5),1);
```

```
68 end
69     % max RUL is adjusted to 190
70     Max_Value_D=190;
71 if size(produced_data{BaseTrain, 1},1 )-(StartBaseTrainsize+
   Kn)>Max_Value_D
72     StartBaseTrainsize=StartBaseTrainsize+((size(
           produced_data{BaseTrain, 1},1 )-(StartBaseTrainsize+
           Kn))-Max_Value_D);
73 end
74     % Produced data to be used
75     produced_data{i,3}=produced_data{BaseTrain, 1}(
           StartBaseTrainsize:StartBaseTrainsize+Kn,:);
76     % True RUL
77     produced_data{i,4}=size(produced_data{BaseTrain, 1},1)-(
           StartBaseTrainsize+Kn);
78     % Baseline training data without trimming
79     produced_data{i,5}=produced_data{BaseTrain, 1}(
           StartBaseTrainsize:end,:);
80 end
81 end
```

Appendix D

D.1 Score Table

Prognostic metrics for secondary data sets - The terms, a b and c, stand for different window sizes used in the moving average filtering.

Dataset & Windows Size	Scoring Func.	FPR	FNR	MAPE	MAE	MSE
DS 1 #a	8280.807	52%	48%	24.639	13.018	410.058
DS 1 #b	7489.328	51%	49%	24.17	12.743	394.345
DS 1 #c	7396.534	49%	51%	23.552	12.398	387.982
DS 1 #mean	7298.195	50%	50%	23.685	12.541	388.8
DS 2 #a	3327.28	53%	47%	23.517	11.776	318.509
DS 2 #b	3313.39	53%	47%	23.485	11.841	319.656
DS 2 #c	3168.63	54%	46%	23.444	11.926	320.839
DS 2 #mean	3104.176	53%	47%	23.17	11.715	312.206
DS 3 #a	4632.998	50%	50%	24.242	12.206	363.185
DS 3 #b	5545.529	52%	48%	23.641	11.977	363.213
DS 3 #c	5143.634	54%	46%	23.241	11.788	352.588
DS 3 #mean	4580.182	53%	47%	23.412	11.854	353.256
DS 4 #a	4854.999	50%	50%	23.75	11.451	313.297
DS 4 #b	4187.894	53%	47%	22.812	11.191	302.759
DS 4 #c	4261.651	51%	49%	22.179	11.149	306.979
DS 4 #mean	4189.981	52%	48%	22.464	11.143	299.881

Dataset & Windows Size	Scoring Func.	FPR	FNR	MAPE	MAE	MSE
DS 5 #a	4434.338	47%	53%	21.379	11.995	347.54
DS 5 #b	3669.566	51%	49%	20.894	11.571	323.214
DS 5 #c	3678.593	52%	48%	20.282	11.432	319.326
DS 5 #mean	3746.6	50%	50%	20.439	11.498	321.969
DS 6 #a	5365.458	49%	51%	25.509	12.738	368.69
DS 6 #b	6621.318	49%	51%	24.165	12.312	365.779
DS 6 #c	6538.401	51%	49%	23.902	12.106	354.481
DS 6 #mean	5779.512	50%	50%	24.216	12.213	354.53
DS 7 #a	5187.46	50%	50%	22.937	12.224	366.564
DS 7 #b	4214.101	52%	48%	22.178	11.929	347.125
DS 7 #c	4588.46	50%	50%	21.686	11.713	346.796
DS 7 #mean	4368.215	51%	49%	21.885	11.817	345.268
DS 8 #a	3863.144	46%	54%	25.109	11.692	313.279
DS 8 #b	3604.684	48%	52%	24.425	11.697	312.476
DS 8 #c	3585.884	47%	53%	24.334	11.704	314.924
DS 8 #mean	3406.429	46%	54%	24.29	11.503	304.79
DS 9 #a	5935.363	49%	51%	23.501	12	362.183
DS 9 #b	5449.336	50%	50%	23.147	11.818	357.478
DS 9 #c	5752.652	49%	51%	23.122	12.011	363.529
DS 9 #mean	5358.706	50%	50%	23.039	11.836	352.054
DS 10 #a	6666.386	51%	49%	24.066	11.956	367.766
DS 10 #b	6607.538	54%	46%	23.153	11.806	371.153
DS 10 #c	7378.55	52%	48%	22.565	11.849	385.884
DS 10 #mean	6523.669	53%	47%	22.812	11.749	366.863
DS 11 #a	5240.953	52%	48%	23.123	12.027	359.788
DS 11 #b	4954.653	52%	48%	22.174	11.646	352.448

Dataset & Windows Size	Scoring Func.	FPR	FNR	MAPE	MAE	MSE
DS 11 #c	5572.072	51%	49%	22.215	11.73	360.785
DS 11 #mean	4485.173	52%	48%	22.061	11.664	348.143
DS 12 #a	2452.792	48%	52%	24.118	11.076	260.71
DS 12 #b	2229.2	49%	51%	23.605	10.932	253.722
DS 12 #c	2137.85	47%	53%	23.27	10.878	250.864
DS 12 #mean	2123.455	49%	51%	23.4	10.86	248.391
DS 13 #a	5094.571	48%	52%	25.376	9.958	266.519
DS 13 #b	5467.651	52%	48%	24.147	9.666	263.74
DS 13 #c	6668.547	50%	50%	23.913	9.709	271.603
DS 13 #mean	5259.011	50%	50%	23.968	9.662	261.662
DS 14 #a	5027.927	46%	54%	25.205	12.613	347.974
DS 14 #b	6117.644	48%	52%	24.757	12.588	351.446
DS 14 #c	5467.947	50%	50%	24.858	12.638	344.062
DS 14 #mean	5238.378	48%	52%	24.517	12.354	336.798
DS 15 #a	4854.999	50%	50%	23.75	11.451	313.297
DS 15 #b	4187.894	53%	47%	22.812	11.191	302.759
DS 15 #c	4261.651	51%	49%	22.179	11.149	306.979
DS 15 #mean	4189.981	52%	48%	22.464	11.143	299.881
DS 16 #a	3155.883	50%	50%	23.617	11.537	289.743
DS 16 #b	2959.411	51%	49%	22.867	11.147	280.844
DS 16 #c	2982.014	50%	50%	23.011	11.366	292.702
DS 16 #mean	2745.895	51%	49%	22.748	11.233	279.712
DS 17 #a	4511.288	50%	50%	23.691	11.24	298.833
DS 17 #b	3867.355	53%	47%	22.705	10.906	282.853
DS 17 #c	4001.487	51%	49%	22.091	10.894	289.533
DS 17 #mean	3896.616	52%	48%	22.379	10.892	282.734

Dataset & Windows Size	Scoring Func.	FPR	FNR	MAPE	MAE	MSE
DS 18 #a	3139.819	45%	55%	22.842	10.929	262.942
DS 18 #b	2194.265	48%	52%	21.779	10.38	241.801
DS 18 #c	2217.07	48%	52%	21.139	10.031	230.706
DS 18 #mean	2189.902	48%	52%	21.601	10.281	236.468
DS 19 #a	3460.168	52%	48%	23.472	12.331	320.672
DS 19 #b	3338.779	53%	47%	22.895	11.938	309.028
DS 19 #c	3330.939	51%	49%	22.213	11.653	303.067
DS 19 #mean	3085.239	51%	49%	22.487	11.808	302.741
DS 20 #a	6377.481	43%	57%	25.231	11.867	341.93
DS 20 #b	6250.569	46%	54%	24.585	11.715	332.783
DS 20 #c	6225.134	45%	55%	24.34	11.616	330.66
DS 20 #mean	5668.858	44%	56%	24.418	11.598	326.571

Table D.1: Prognostic metrics for secondary data sets - The terms, a b and c, stand for different window sizes used in the moving average filtering