Statistical Hierarchical Modelling for Industrial Collaborative Prognosis

Maharshi Harshadbhai Dhada

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Churchill College March 2022

"I love deadlines. I love the wooshing sound they make as they pass by."
- Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Maharshi Harshadbhai Dhada
March 2022

“Statistical Hierarchical Modelling for Industrial Collaborative Prognosis”

by

Maharshi Harshadbhai Dhada

Abstract

Recent advancements in computing, telecommunications, and metrology have propelled data-driven decision making across the industries. Industrial health management in particular has been increasingly reliant on Machine Learning techniques for data-driven prognosis as modern assets are exhaustively monitored by their embedded sensors.

Data-driven prognosis constitute the bedrock for the emerging highly-flexible predictive maintenance policies. The original equipment owners and customers also mutually enjoy its cost benefits via servitisation for example, where the customers pay for asset uptimes rather than ownerships. Such business offerings are possible only due to the recent advent of distributed control systems and real-time prognosis.

However, the diversity in asset operating conditions results in non-ergodicity which is often a challenge while modelling the asset fleet data. A fleet-wide model trained by pooling the data from all the assets is associated with a high bias, whereas the independent assets-specific models are associated with high variance for the assets with sparse data. Thankfully there exist similarities due to age, upkeep, manufacturing processes, etc. across the assets that enable learning possibilities within the asset fleet, via collaborative prognosis.

This thesis proposes, and demonstrates, that statistical hierarchical modelling is a systematic technique for collaborative prognosis and also for anomaly detection in the asset condition data. Hierarchical models presented herewith extend the independent models by formulating distributions at multiple levels, such that the parameters of the lower level assets-specific models are commonly sampled from the corresponding higher level distributions. This encourages learning for the assets with sparse data as they inherit prior information about the operations from the other similar assets comprising the fleet.

It is concluded that, for prognosis and anomaly detection, the hierarchical models outperform the independent and the fleet-wide models in terms of accuracy and variance for the assets with sparse data. Hierarchical models model the asset fleets in their natural order and also enable manual intervention for modelling the asset similarities via the higher level distributions. As data is accumulated along the asset operations, both hierarchical and independent models converge similarly. The conclusions are also supported by a case study presented for an industrial fleet of long-haulage trucks.

Acknowledgements

I dedicate this thesis firstly to my parents- for your immense love, cheer, and putting my interests above all. I wouldn't have been in and through Cambridge without you, as even miles away from home talking to you would make everything achievable.

Secondly to my supervisor Professor Ajith Parlikad. In addition to your guidance and flexibility, I would forever be grateful to you for believing in me. Ajith has undoubtedly been the best supervisor I could ask for.

To Professor Mark Girolami and Lawrence for I have learnt a lot from you, and your inputs have been instrumental in the course of my PhD.

To my friends Suraj and Fadil who have been with me throughout the PhD journey. With playing Fifa, squash, golf, travelling, queueing for Wimbledon, football matches, cooking, jamming, movie screenings, etc. you have also always been inspirational.

To my friends and DIAL-mates Manuel, Georgios, Manu, Adolfo, Nicola, and Yaniv for being with me in times of celebration and hustle! Also to Becky, Gokcen, Anandrup, Marco, Adria, Amit, Vinod, Jorge, Xiang, Tim, Zhenglin, Olof, Tony, Mukesh, and Bhupesh who made the workplace more than just a workplace.

To Jazmin, Kunal, Pranjal, Harneet, Kirthana, Arthi, Gurashish, Partha, Dhruv, Asmita, Omkar, Ramit, Arushi, Shujath, Rishi, Muntazir, and all the life-long friendships I have made via Cambridge.

I would never forget the Darwin College Bar (DarBar), which I can relate my most funny Cambridge memories to. Thanks to my friends Georgios, Adolfo, Sai, Georgia, Christina, and the DarBar staff for making DarBar, well, DarBar.

The Covid-19 pandemic hit half-way through my PhD second year, and Cambridge felt no less than home thanks to the Churchill College, my friends Asmita, Lambed, Trisha, and my housemates in 36A (Emma, Luke, Jeremy, and Tamara) - for hearing me out, working out hobbies together, cycling trips, and most importantly, tolerating me at times.

My time in Cambridge has been endlessly fascinating and definitely constitute the best years of my life (yet). I learnt rowing, gliding, salsa, played tennis in some of the best courts, ran in sub-zero temperatures, enjoyed football matches in the pubs and the stadiums, cycled on the best routes, travelled my heart out, found myself, and worked with the smartest people.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Unprecedented advancements in the fields of computing, telecommunications, and metrology over recent years have revolutionised various sectors. The industrial sector in particular is undergoing the fourth industrial revolution with technologies such as the Internet of Things, Machine Learning, Virtual Reality, etc. now widespread [178]. Combined benefits of the low-cost computations, high-speed communication, and a plethora of sensing technologies specifically are propelling the increasing applications of data-driven techniques for optimising the industrial operations [178]. This thesis contributes towards one such application, of data-driven prognosis for industrial asset fleets.

Data-driven prognosis involves Machine Learning techniques to model the relationships between the operational data from the assets such as their condition, age, specifications, human inputs, etc. and their remaining useful lives (RULs) [161]. Apart from adding to the ease of use, data-driven prognosis also forms the bedrock for the emerging maintenance planning strategies such as predictive maintenance [178]. It is therefore critical for asset management in general that aims at maximising the overall life-cycle value of the industrial assets [67].

Unfortunately the assets often do not posses sufficient failure data by themselves for training the prognosis models as the operational data are incrementally collected throughout the asset life-cycles [133, 23]. Moreover, the data pertaining to certain failure types or unusual operating conditions are extremely rare [139]. It poses a challenge as failure data are necessary for data-driven prognosis.

Unavailability of failure data is particularly challenging because the assets operate in diverse conditions and environments, rendering the asset fleet a statistically non-ergodic system [139, 158]. Therefore, the prognosis models corresponding to the assets with sparse training data must identify and learn from the failures observed in other similar assets comprising the fleet to optimise failure predictions [158].

Collaborative prognosis is a concept that aims at enabling the models corresponding to the assets with sparse training data to identify similar other assets in the fleet and enhance their locally available information [158]. However, a systematic technique for enabling collaborative prognosis is a critical research gap.

This chapter introduces the reader to data-driven prognosis, the research questions addressed in this thesis, and also presents a structure of this thesis. The following chapter is structured as: Section 1.1 introduces the reader to data-driven prognosis, followed by a brief discussion about the state-of-the-art collaborative prognosis and problem formulation. The significance of anomaly detection for prognosis is also explained in Section 1.1. Section 1.2 outlines the research questions and research objectives addressed in this thesis. Section 1.3 describes the research methodology followed to address and achieve the research questions and objectives respectively. The structure of this thesis is provided in Section 1.4 in the form of descriptions of the following chapters and mentioning the corresponding research objectives achieved in each of them.

1.1 Conceptual Introduction

This section introduces the reader to data-driven prognosis in Section 1.1.1, and provides a a brief discussion about the state-of-the-art collaborative prognosis for problem formulation in Section 1.1.2. The significance of anomaly detection for prognosis is explained in Section 1.1.3

1.1.1 Introduction to Data-Driven Industrial Prognosis

An asset's life-cycle comprises of *plan, acquire, use, maintain,* and *dispose* phases, out of which majority of their lives are spent by the assets in *use* and *maintain* phases [116]. Resultantly, industries also bear the maximum cost while operating and maintaining their assets.

Maintenance planning in particular is necessary for the upkeep of the assets and is an optimisation problem of *preventing asset failures with minimal maintenance interventions*. This is graphically represented in Figure 1.1[1].

In Figure 1.1, preventive maintenances refer to the regular checkups that *prevent* failures, whereas corrective maintenances refer to the emergency responses to the unexpected asset failures. The total maintenance cost is minimised by strategically planning the preventive

[1] sourced from https://risktec.tuv.com/risktec-knowledge-bank/asset-integrity-management/emit-optimisation-getting-more-out-of-existing-equipment-for-less/

maintenance interventions so that they lie in the optimal maintenance zone depicted in Figure 1.1. The location of the optimal maintenance zone depends on the application and the ratio of the preventive to the corrective maintenance costs, along with the definition of the costs from the application's perspective [178, 180].

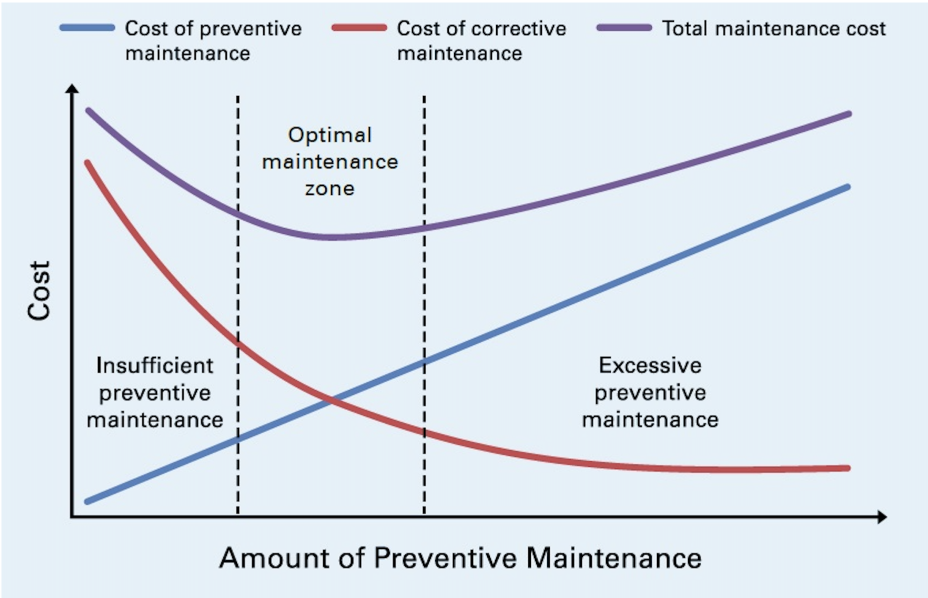


Fig. 1.1 Graph describing optimal maintenance planning.

Industrial managers popularly rely on asset reliability estimations for maintenance planning, where asset reliability refers to *the probability that the asset will perform its required function under the given conditions for the stated interval of time* [133, 174]. Instantaneous asset reliability is mathematically expressed as the probability of success, or non-failure, $R(t)$ at time interval $[t, t+dt]$ as the time interval dt tends to zero [133]. Asset reliability also loosely translates to asset health such that assets in good health are characterised by high reliabilities or for the case of prognosis, longer RULs.

Industrial prognosis is dedicated to predicting an asset's RUL, often via estimating its reliability. Data-driven prognosis has been gaining increasing popularity in the industries, thanks to the computational and metrological advancements [161]. Modern assets are embedded with sensors that monitor a variety of internal and external condition parameters in real-time [161]. These include parameters such as temperature, vibrations, etc. that resemble the asset's health at the given instance. The time-series of condition data ranging from the start of the asset deteriorations until their failures, called failure trajectories, are used to train models using Machine Learning techniques that predict their future occurrences for the given asset's condition [161]. It should be noted that the asset condition data comprise of, but are not limited to, the time-series of sensor measurements.

Data-driven prognosis forms the bedrock for the state-of-the-art predictive maintenance policies, characterised by real-time asset failure predictions and maintenance interventions on as-needed basis [157]. Predictive maintenance has has unlocked unmatched cost savings for the original equipment manufacturers (OEMs), and at the same time new business offerings like servitisation for the customers [129].

1.1.2 Problem Formulation

The failure trajectories are often characterised by the diversity in asset operating conditions due to age, upkeep, operations, manufacturing process, etc [139, 158]. Nonetheless, the industries either commonly model the asset failures across the fleet, or rely on independent models that are trained using the asset-specific data only. Both these approaches are associated with drawbacks because often a fleet-wide model trained using all the data pooled together is bound to be associated with a high bias, whereas independent assets-specific models are associated with high variances especially if an asset has sparse number of failure occurrences [104, 155]. An over-parameterised fleet-wide model can potentially address the problem of high bias but at the same time require a larger training dataset, or lead to problems of overfitting on the other hand [8, 27]. As such, a systematic technique of modelling asset condition data arising from a fleet of assets is much needed.

Thankfully the factors leading to non-ergodicity in the fleet are systematic, in the sense similarities often exist across the assets. The literature presents evidence that collaborative learning can be achieved for data-driven prognosis, such that the lack of failures to train an independent model for the asset with sparse data is mitigated by identifying and learning from similar other assets in the fleet. The concept of collaborative prognosis involves identifying the sub-fleets of similarly deteriorating assets and exchanging the information within these sub-fleets. Collaborative prognosis is especially useful while predicting failures in the early asset operations or for the assets operating in dynamic environments such that the data pertaining to a given environment or individual assets are not sufficient for prognosis [158, 7, 156, 135, 157]. However, a critical hinderance to enable collaborative learning in the industries is the lack of a systematic technique to share information within the sub-fleets of assets undergoing similar deterioration.

Problem Statement: To present a systematic technique for modelling the failure data in asset fleets, and enabling collaborative prognosis for the assets with sparse data.

This thesis proposes and demonstrates that the statistical hierarchical modelling is a systematic technique to enable collaborative prognosis for those assets with sparse data, and in the presence of similarly deteriorating data-rich assets in the fleet. Hierarchical models are characterised by shared higher level distributions for the parameters of the independent models of similar assets [52, 53]. In such a hierarchy, the asset data are sampled from the independent models, and the parameters of the independent models are in turn sampled from the corresponding higher level distributions. When an asset does not have sufficient data, the higher level distributions incorporate the failure data from other similar assets to learn prior information for the asset with sparse data. The higher level distributions in a hierarchical model represent the general behaviour of similar assets, whereas the individual asset behaviours are described by the parameters sampled from corresponding higher level distributions [52, 53].

1.1.3 Anomaly Detection as a part of Prognosis

Furthermore, it should be noted that the failures are often not incipient since the start of asset operations. The incipience of the failures manifest as anomalies in the time-series of asset condition data while they are operating [134]. This makes anomaly detection crucial for data-driven prognosis for two reasons:

1. Anomaly detection acts as a trigger for activating the prognosis algorithms for an asset under operation. An ideal anomaly detection algorithm instantaneously identifies deviations in the time-series of asset condition data in real time, and activates the prognosis algorithm. An inefficient anomaly detection algorithm instead could let anomalies go undetected, or flag many anomalies that turn out to be benign and not require any intervention [88].

2. Anomaly detection is essential for identifying failure trajectories corresponding to the observed failures in the asset fleet. A failure trajectory ranges from the point where the anomaly was detected, marking the incipience of failure, until the asset failure was observed. Since historical failure trajectories constitute the training dataset for prognosis, learning capabilities of the prognosis models primarily depend on accurate anomaly detection [134].

Most industries rely on rule based systems for anomaly detection. These comprise of preset warnings and trip limits on the sensor measurements [199, 160]. Force tripping an asset often results in avoidable production losses if a planned maintenance was carried out earlier. Moreover, the warning-trip systems are inherently non-responsive as an asset, for example, could be operating well within the limits but also be deviating from its normal behaviour. This deviation would not be flagged by a warning-trip system until sensor measurements exceed the preset limits, which could be too late to plan maintenance interventions. This drawback is explained in Figure 1.2.

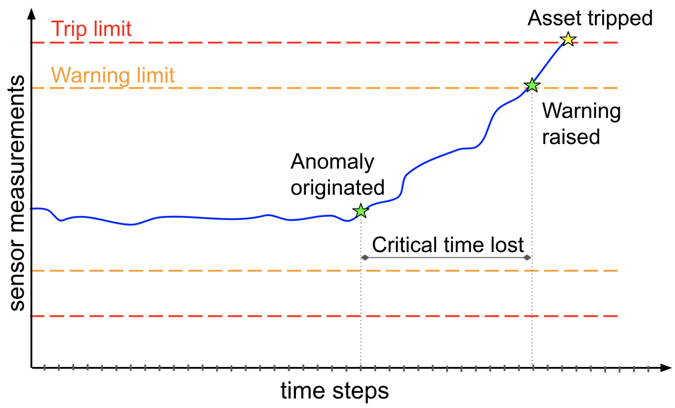


Fig. 1.2 Schematic explanation of the drawback of conventional warning-trip systems.

Statistical classifiers are also used for anomaly detection in the time-series of asset condition data. Statistical classifiers posit that the condition monitoring data generated during normal asset operations can be described using underlying distributions. Assuming that an asset commences operating in normal condition, the underlying density function $p(\theta)$, θ being its parameters, can be estimated to model that asset's normal operation data. Upcoming anomalies in asset operations cause a change in system dynamics, and induce deviation from its estimated density function. Statistical tests are used to evaluate if a newly recorded data point is significantly different to be deemed anomalous [88, 151].

But independent modelling of assets, that is using an independent statistical classifier for each asset in the fleet, is accompanied with distribution instabilities. Depending on the variance in asset data, distribution parameters would not be stable until certain amount of data describing the asset's working regime is obtained. Moreover, collective modelling of the fleet wide data is challenging given the statistically diverse nature of the asset operations [158, 30].

The problem of anomaly detection is therefore critical for data-driven prognosis, and it is discussed here that a similar problem as that for prognosis exists for anomaly detection for asset fleets. Statistical hierarchical modelling is proposed in this thesis also as a technique to enable collaborative learning for anomaly detection for assets with sparse data. The following section outlines the research questions, and the research objectives targeted and achieved in this thesis to address the identified research problem.

1.2 Research Questions and Objectives

This section outlines the research questions, and the research objectives targeted and achieved in this thesis.

1.2.1 Research Questions

The following research questions are outlined to address the research problem described in Section 1.1.2:

- **Question 1:** How to model the asset fleet data systematically to enable collaborative prognosis and anomaly detection? This research question aims at addressing a crucial research gap in collaborative prognosis literature for a systematic technique to model and enable collaborative learning in a fleet comprising of clusters of similarly operating assets. Statistical hierarchical modelling is proposed in this thesis as a solution for enabling collaborative prognosis.

- **Question 2:** How effective are the statistical hierarchical models for collaborative prognosis and anomaly detection? This research question aims at exploring the advantages and the limitations of using statistical hierarchical models for collaborative prognosis. It is demonstrated, using simulated and industrial datasets, that the hierarchical models outperform the independent and the fleet-wide models for prognosis and anomaly detection in terms of accuracy and variance for the assets with sparse data.

1.2.2 Research Objectives

The research questions defined in Section 1.2.1 are further distilled into specific objectives, which if achieved would address the research questions.

- **Research Objective 1:** Explore the existing collaborative learning techniques in industrial health management and other applications involving distributed data similar to that of an asset fleet.

- **Research Objective 2:** Propose a technique for collaborative anomaly detection.

- **Research Objective 3:** Analyse the proposed statistical hierarchical model for anomaly detection.

- **Research Objective 4:** Propose a technique for collaborative prognosis.

- **Research Objective 5:** Analyse the proposed statistical hierarchical model for collaborative prognosis.

- **Research Objective 6:** Analyse the applicability of the proposed statistical hierarchical model for collaborative prognosis of an industrial fleet.

1.3 Research Methodology

In order to address the research problems and achieve the research objectives, the research presented in this thesis was conducted in six stages. It was ensured that while being theoretically sound the proposed solutions were also feasible and practical for the industries. Figure 1.3 summarises the stages of the research methodology followed herewith.

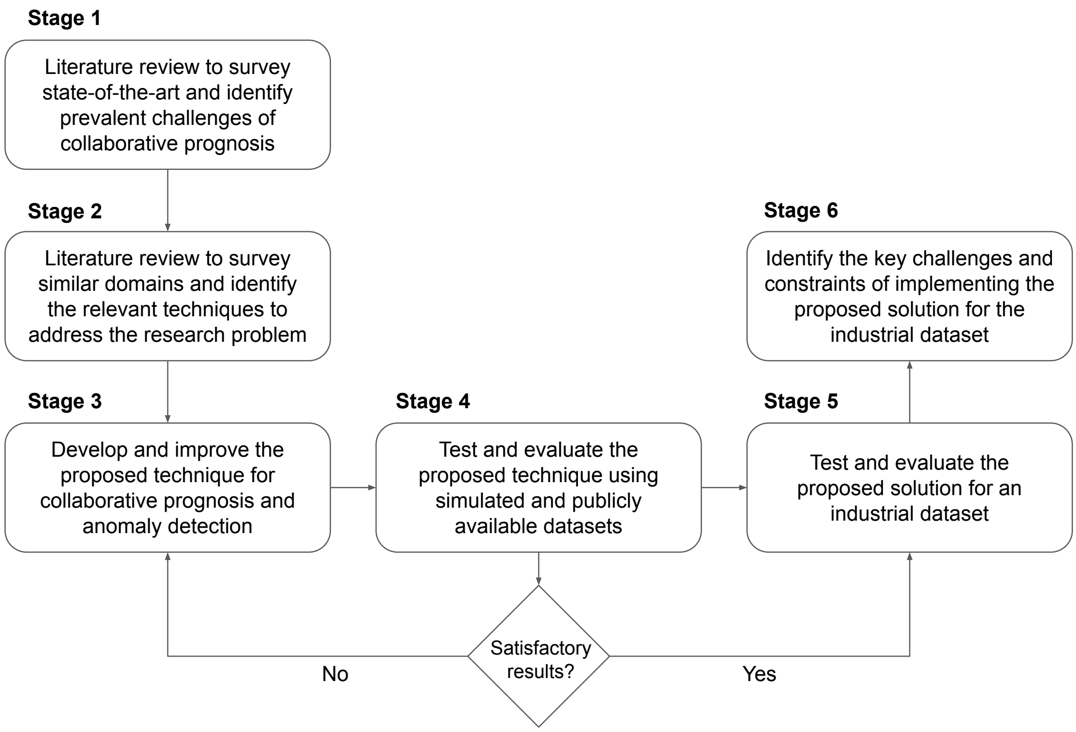


Fig. 1.3 A summary of the stages followed in the research methodology for this thesis

The initial stages 1 and 2 involved (i) studying the concepts of maintenance planning and the state-of-the-art in the industrial prognosis, and (ii) identifying the relevant techniques from similar other domains to address the problem of collaborative prognosis in asset fleets. It was in these phases that the need for a systematic technique for collaborative prognosis was identified, and concluded that statistical hierarchical modelling is a relevant solution for enabling the same. These stages of the research methodology also involved ensuring the relevance of this problem for the industries via discussions with the experts and surveying the asset condition data in the telecommunications, power generation, railways, and automotive industries.

Stages 3 and 4 involved the testing and development phases of the proposed solution for collaborative prognosis. Relevant datasets available in public domain were identified for evaluation of the proposed technique, and where not publicly available it was ensured that the self-generated data were realistic. Using simulated datasets ensured that the efforts were concentrated on analysing and improving the performance of the proposed techniques rather than on preprocessing the data which is often the case with the real world datasets. The proposed hierarchical models for collaborative anomaly detection and prognosis were then separately tested or modified using the simulated datasets in a feedback loop presented in Figure 1.3.

Finally in stages 5 and 6, the proposed hierarchical model for collaborative prognosis was implemented for an industrial dataset. To that end, data from a large fleet of long-haulage trucks was obtained, where the trucks were monitored across a long enough timespan ranging since the start of their operations until the failure of the targeted components. As a result, key challenges and constraints while implementing the hierarchical model for an industrial dataset were identified. The implementation of the proposed hierarchical model for collaborative prognosis for an industrial dataset is presented in this thesis as a case study.

1.4 Thesis Outline

This section describes the structure of the following thesis in the form of brief summaries of the following chapters and also mentioning the corresponding research objectives addressed in the chapters, in the following points. The chapter summaries along with the research questions and objectives answered/ achieved in the corresponding chapters are also presented in Figure 1.4.

- **Chapter 2: Research Background** The aim of this chapter is to highlight specific challenges and research gaps for collaborative prognosis, based on a review of literature of data-driven prognosis of asset fleets, collaborative prognosis, and anomaly detection. Concepts of Federated Learning and statistical hierarchical models are also briefly discussed to provide the foundational concepts for the algorithms presented in this thesis. This chapter addresses the research objective 1.

- **Chapter 3: Statistical Hierarchical Model for Collaborative Anomaly Detection** Anomaly detection is critical for triggering the prognosis algorithms, and also to extract the failure trajectories from the historical data. A hierarchical model for anomaly detection capable of identifying the sub-fleets of similar assets and collaborative learning is presented in this chapter. Analytical solution to update the model parameters is derived such that the assets with sparse data can identify and learn from other similar assets. The experiments are conducted using a simulated data resembling an asset fleet and compare independent models, fleet-wide model, and a hierarchical model for varying proportions of sparse-data assets. Experiments are also conducted to analyse the effect of increasing amount of data in the sparse-data assets. This chapter addresses the research objectives 2 and 3.

- **Chapter 4: Collaborative Prognosis using a Statistical Hierarchical Model** Chapter 4 proposes a technique for identifying the sub-fleets of similarly deteriorating assets using the observed failure trajectories, followed by modelling the times-to-failures observed across the fleet using a statistical hierarchical model of Weibull density functions. The experiments are conducted with a simulated fleet of turbofans, and compare the performance of the independent, fleet-wide, and hierarchical models with increasing number of failures observed in a cluster, and also the effect of higher level models on modelling for the case of the hierarchical model. This chapter also presents the procedure for implementing the hierarchical model for real-time collaborative prognosis and analyses the effect of clustering on the prediction performance. Chapter 4 addresses the research objectives 4 and 5.

- **Chapter 5: Industrial Case Study: Modelling Failures in a Fleet of Heavy-duty Trucks** This chapter discusses a case study where the proposed hierarchical Weibull model is implemented for modelling the times-to-failures in a fleet of heavy-duty trucks. Apart from proving the necessity of collaborative prognosis, this case study enables identifying the major challenges faced while implementing the statistical hierarchical models for an industrial dataset. This chapter addresses the research objective 6.

- **Chapter 6: Conclusion and Future Research Directions** This chapter presents the general conclusions to this thesis along with a summary of the academic contributions, limitations of the proposed techniques, and the future research directions.

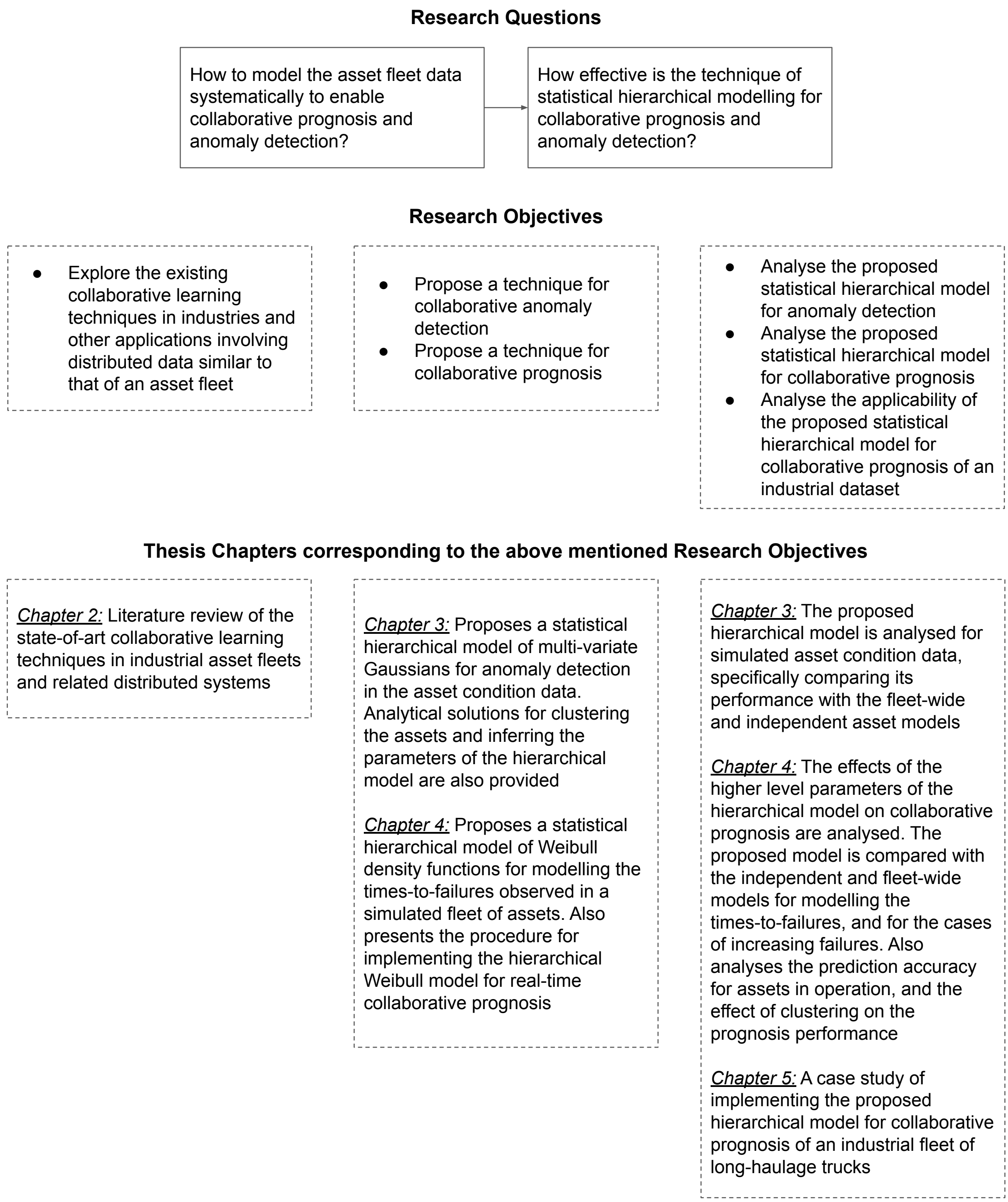


Fig. 1.4 A summary of Research Questions addressed in this thesis. Research Objectives and the corresponding chapters are also mentioned.

Chapter 2

Research Background

This chapter presents a literature review of data-driven prognosis, collaborative prognosis, and anomaly detection in the industrial asset management domain to highlight specific challenges and research gaps for collaborative prognosis. The foundational concepts for the algorithms presented in this thesis and popular applications of statistical hierarchical modelling are also discussed. Moreover, other popular collaborative learning techniques prevalent in the industries, or targeting similar problems are discussed towards the end to compare with the technique of statistical hierarchical modelling. This chapter addresses research objective 2 outlined in Chapter 1.

The following chapter is structured as: Section 2.1 introduces the concept of maintenance planning and reliability, and reviews the literature pertaining to data-driven prognosis, collaborative prognosis, and anomaly detection in the industries. Section 2.2 provides a brief discussion of various collaborative learning techniques, and motivates the choice of statistical hierarchical modelling as a suitable technique for modelling asset fleet data, and thus being a systematic technique for collaborative learning. Section 2.3 provides an example implementation of a hierarchical linear regression model for asset fleet data in addition to discussing various engineering applications of the hierarchical models, and finally the conclusions from this chapter are summarised in Section 2.4.

It should be noted that asset management in the context of this thesis refers to the physical asset management, which is defined as "the practice of managing the entire life cycle (design, construction, commissioning, operating, maintaining, repairing, modifying, replacing and decommissioning/disposal) of physical and infrastructure assets such as structures, production and service plant, power, water and waste treatment facilities, distribution networks, transport systems, buildings and other physical assets" [67]. In other words, asset management aims at *maximising the overall life-cycle value of the assets*. International Standard for Organization describes the formal terminologies, requirements, and guidance for asset management, and also the practices for implementing, maintaining and improving an asset management system in their ISO 55000 series [67]. The updated ISO 55001 series further incorporates the cloud-computing, and internet of things (IoT) technologies into the framework of asset management [58].

2.1 Data-driven Prognosis

2.1.1 Introduction to Data-driven Prognosis

This section introduces the concept of maintenance planning and reliability, and reviews the literature pertaining to anomaly detection, data-driven prognosis, and collaborative prognosis in asset management.

The industries bear maximum costs from the operations and maintenance phases of an asset's life-cycle [116]. Maintenance planning in particular is critical for asset management and has substantially evolved over the years, through reactive, preventive, predictive, and now the emerging concept of prescriptive maintenance strategies [5].

Manufacturers design for maintainability to ensure effective, safe and economic maintenance interventions during the asset's life-cycle [176]. Nonetheless, the maintenance interventions are associated with labour, downtime, and spare parts costs, and are conventionally classified as preventive or corrective [180]. Preventive maintenances are the regular checkups and interventions that *prevent* imminent failures, whereas corrective maintenances are the emergency responses to unexpected asset failures [180].

Corrective maintenances are often significantly more expensive than the preventive maintenances. This is because of the additional cost of reduced output or downtime resulting from the unplanned procurement of the repair equipment, spare parts, and crew required for corrective maintenances [180]. E.g. An asset's downtime in an automotive industry can result in a loss of upto $1,000 per min [167]. A failing wind turbine bearing can result in a maintenance operation multiple times higher than the actual cost of the component [96]. For safety critical assets, such as aircrafts, the costs of an unexpected failure potentially also includes intangible social losses.

Industries aim at minimising the asset failures, and in turn the corrective maintenance costs, via regular preventive maintenances. But at the same time, over-maintenance of the healthy assets should also be avoided to optimise the total maintenance costs [180]. Ideally, maintenance planning distills to an optimisation problem of *minimising the total maintenance* *cost by strategically planning the preventive maintenances*, which is graphically represented in Figure 2.1[1].

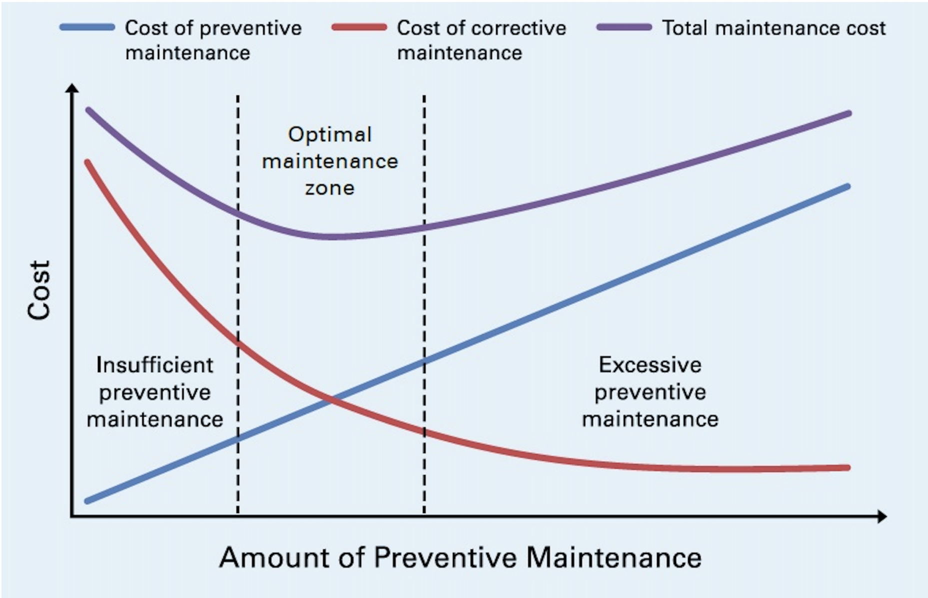


Fig. 2.1 Graph describing optimal maintenance planning.

An extreme strategy, represented by the *Excessive preventive maintenance* region in Figure 2.1 corresponds to entirely avoiding the asset failures by conducting preventive maintenances with high frequency. This causes a situation where the healthy assets also undergo maintenance resulting in increased costs to the owners. The other extreme strategy, represented in *Insufficient preventive maintenance* region in Figure 2.1, corresponds to not conducting any preventive maintenances, leading to widespread asset failures. An ideal maintenance plan must lie between these two, and is represented as the *Optimal maintenance zone* of the plot shown in Figure 2.1. In the optimal maintenance zone, the asset failures may not entirely prevented but the total maintenance cost is minimised.

The location of the optimal maintenance zone depends on the application and the ratio of the preventive to the corrective maintenance costs, along with the definition of the costs from the application's perspective. Industrial managers popularly rely on reliability-centered maintenance policies, to mitigate the asset failures by maximising asset reliabilities via optimally timed interventions [15].

Asset reliability is defined as the probability that an asset will perform its required function under given conditions for a stated interval of time. Reliability is mathematically

[1]This figure is sourced from https://risktec.tuv.com/risktec-knowledge-bank/asset-integrity-management/emit-optimisation-getting-more-out-of-existing-equipment-for-less/

expressed as the probability of success, or non-failure, $R(t)$ at time interval $[t, t+dt]$ as the interval of time dt tends towards zero [133, 174].

Industrial prognosis is dedicated to predicting an asset's reliability by estimating its remaining useful life (RUL) or the probability of the asset's failure in a future time horizon, given the asset's operating conditions and historical failures [197]. In the context of prognosis, asset reliability also loosely translates to asset health such that assets in good health are characterised by high reliabilities or longer RULs. The broader *Prognostics and Health Management (PHM)* framework links prognosis with the measures to mitigate the impact of the predicted failures via measures such as maintenance activities, load sharing, human intervention, etc. [197].

Asset prognosis is classified as physics-based prognosis or data-driven prognosis, although hybrid prognosis techniques are also emerging in the recent literature such as in [152, 127]. Physics-based prognosis relies on expert knowledge and physical laws of failure to predict an asset's RUL [152, 174]. Industrial experts study the failure modes in laboratories via accelerated life testing, to functionally formulate the relationship between an asset's operating condition and its estimated RUL. An example of physics-based prognosis model is the crack growth equation [2]. Physics-based prognosis is unfortunately limited to only certain components/ failure modes for which sufficient theory exists.

Nevertheless, modern assets are characterised by embedded sensors that continuously record a variety of internal and external operating parameters [152, 60]. Time-series of asset condition data such as vibration, temperature, creep, or even automatic alarms representing the asset health histories are now, thanks to the declining costs of sensors, standard across the industries. The increase in the availabilities of asset condition data in the industries have propelled the growth of data-driven prognosis over the recent years [161, 80, 152]. The time-series of condition data ranging from the start of the asset deteriorations until their failures, called failure trajectories, are used to train models using Machine Learning techniques that predict their future occurrences for the given asset's condition [161, 152].

In fact, the potential of using computational models to enable real-time prognosis has been known since the 1980's but their application was hindered by lack of data and expensive computations [19]. The analytics pipeline for data-driven prognosis involves (1) identifying a failure type for given operating conditions of assets, (2) training prediction model using historical trajectories of that failure, and (3) implementing the optimised prediction model in real time [98, 161]. For the applications where prognosis is treated a regression problem, the independent variables comprise of the asset health indicators and the dependent variable is the corresponding time to failure. A classification problem can also be formulated for prognosis, where a given set of health indicators corresponds to either failure or non-failure data. It should be noted that the asset condition data comprise of, but are not limited to, the time-series of sensor measurements [152].

The literature presents exceeding examples where almost all categories of algorithms- fuzzy logic, neural networks, bayesian learning, decision trees, support vector regression, etc. have been implemented for prognostics, each with their own benefits and limitations [152]. the review [38] shows the common algorithms used for different machine components. In another review [99], the authors have compiled the different types of data from the assets and matched them with the suitable prognostics algorithms. An example comparison of neural networks, gaussian process regression, and relevance vector machine in terms of their rate of convergence and accuracy for a common dataset is presented in [56]. It can be concluded from these reviews that the data-driven prognosis is widespread across several industries, mostly due to its ease of applications. The choice of prognosis algorithm is often dependant on the component, and the nature of its condition data. For example, [192] introduce Random Forests for regression, and show that they are more accurate than feed forward back propagation neural networks, and SVMs for predicting failures in milling cutters.

Feature extraction and preprocessing have also received popular attention, primarily aiming at reducing the dimensionality of the condition data, while retaining the prognosis ability. Several instances can be found in the literature where the expert knowledge and data science techniques are used to achieve this. For example, the authors in [181] developed feature selection algorithms based on Fast Fourier-Transform which reduce the need for domain expertise while selecting the features. In yet another paper [91], authors propose a feature extraction and selection from the non-trending data. They do this by first de-noising the data using discrete wavelet transform, and subsequently selecting the features using an autoregressive model to identify the relevant features. [74] is a recent publication where the authors rank the features in the raw dataset for the compressor failure modes detection.

Unlike physics-based prognosis, data-driven prognosis significantly reduces the need for expert knowledge or the theory of physical laws of the failure [152]. Moreover, real-time prognosis of the operating assets is possible via data-driven failure prediction models. Data-driven prognosis therefore forms the bedrock for the state-of-the-art predictive maintenance policies, characterised by real-time asset failure predictions and maintenance interventions on as-needed basis [173, 66, 112].

Predictive maintenance has led to unmatched cost savings for the original equipment manufacturers (OEMs) [173, 66, 112]. Especially with advent of remote monitoring and distributed control of the industrial fleets, predictive maintenance have enabled business offerings that only rested in theory until the recent years. Servitisation is one such example where the consumers pay for the asset usages rather than asset ownerships [60]. Servitisation was conceptualised in 1988, but the early offerings of servitisation were only seen in 2013 by Rolls Royce [182, 166, 103, 61]. A large number of firms today rely on substitution “pay-per-use” services that replace the purchase of the product with purchase of the service [16, 119]. Such services are heavily dependent on reducing the unplanned downtime of the assets controlled by the original equipment manufacturer, and therefore on effective predictive maintenance. As a critical precursor to the modern maintenance planning strategies, accurate prognosis can significantly boost the efficiency of an industrial system [98, 81].

2.1.2 Data-driven Prognosis of Asset Fleets

This thesis focuses on data-driven prognosis of a ‘fleet’ of assets. The term *fleet* was originally used to refer to a group of floating vessels but the modern usage covers a whole range of assets including ships, aircrafts, trains to drivetrains, electric transformers, lifts in a building, or machines in a factory [145].

It should be noted that the performance of data driven prognosis relies heavily on the historical failure data used for training the prediction models [152]. An asset fleet however is often characterised by the diversity in operating conditions, model types, or the presence of multiple failure modes. In other words, an asset fleet is a non-ergodic system where every asset has its own unique operating and deteriorating pattern. Prognosis is therefore most accurate if the prediction model learns from a single asset’s failure data only [158]. However, a single asset would need to fail a certain number of times so that necessary training data is available [152, 158]. This is especially true for the high reliability assets, that might not posses the failure trajectories necessary for training a prediction model [158]. Scaling up the algorithms developed for individual asset prognosis to a fleet of assets therefore poses statistical challenges, as the data is *horizontally* distributed across the assets.

Multiple industrial assets are independent, but not identical in statistical sense. Yet, their Independent and Identically Distributed (IID) natures are assumed on several occasions for the ease of modelling. E.g. In [142] authors use the fleet-wide statistics and the fault history to set thresholds on the operating condition, based on which the maintenance activities are planned. [188] use a similarity-based approach assuming the fleet comprises of identical units. [114] developed models which are common for the whole fleet but subsequently, which however undergo changes based on the operating conditions of a single asset. In [153], authors apply the maintenance solution from one system in the fleet to another similar asset. Such approaches are ineffective especially for the asset fleets where the comprising assets are customised, or where the operating conditions of the assets vary widely across the fleet.

For the modern industrial automation almost entirely relying on a series of decision making algorithms, such oblivion to the statistically heterogeneous nature of industrial data poses ever greater risk [95]. For example, a maintenance planning procedure comprises of anomaly detection, failure prediction, maintenance planning, and finally the resource allocation stages [95]. In such a serial dependency, inefficiencies or inaccuracies of an algorithm governing any of these steps perpetuate through the control pipeline and deteriorate the overall efficiency of the system.

Applied mathematicians have stressed on understanding the heterogeneous nature of the industrial assets since 1967. [29] proposed the use of a simple statistical trend test to quantify the evolving reliability of independent industrial assets. The underlying argument was that a single poisson process model could not describe the times between failures occurring in multiple independent assets. [28] further highlighted the importance of understanding inter-asset heterogeneity with an illustration of "happy", "noncommittal", or "sad" assets, corresponding to increasing, constant, or decreasing times between failures respectively. [28] showed that using the trend test proposed by [29] followed by a non-homogeneous poisson processes model, independent industrial assets could be described significantly more accurately.

The algorithms developed for individual assets can naively be deployed for a fleet in either of the two frameworks. In one framework, the assets can be modelled independently by training a prognosis models for each asset in the fleet. In this case the prognosis models would require a large amount of failures to make confident predictions. On the other hand, all assets in the fleet can be jointly modelled using a single model that was used for a single asset. In this case, there is abundance of data as it is accumulated from the entire fleet. But the uncertainty would be much higher than the asymptotic uncertainty of the independent models because the fleet is a mixture of processes [8].

Nevertheless, similarities often arise among the assets comprising an industrial fleet. For example, several levels of similarities may arise in the automobiles, such as the roads, the manufacturer, the chassis design, the driving style, the operation served, etc. In this context, collaborative prognosis is an emerging concept that aims at identifying sub-fleets of similarly deteriorating assets (referred herewith as *clusters* of assets) and enable learning across these sub-fleets (or *clusters*) to improve the overall prognosis performance [158, 7, 156, 135, 157]. This thesis proposes that technique of statistical hierarchical modelling can be suitably used to achieve collaborative prognosis in asset fleets.

2.1.3 Anomaly Detection in Industrial Assets

This section discusses that the problem of anomaly detection is critical for data-driven prognosis, and also that a similar problem as that for data-driven prognosis exists for anomaly detection for asset fleets. Statistical hierarchical modelling is proposed in this thesis also as a technique to enable collaborative learning for anomaly detection for assets with sparse data.

Researchers often treat anomaly detection in asset operations as a one-class time series classification problem [88]. Traditional applications of anomaly detection targeted system diagnostics involving fault identification and classification, with the recent years seeing an increase in online anomaly detection for the asset condition data. This thesis focuses specifically on the statistical classifiers for online anomaly detection, which have been proposed by several researchers for anomaly detection in the data originating from gas turbine combustors, cooling fans, and general performance monitoring due to their versatility [88, 14, 195, 84].

Statistical classifiers posit that the condition monitoring data generated during normal asset operations can be described using underlying distributions. Assuming that an asset commences operating in normal condition, the underlying density function $p(\theta)$, θ being its parameters, can be estimated to model that asset's normal operation data. Upcoming anomalies in asset operations cause a change in system dynamics, and induce deviations from its estimated density function. Statistical tests are used to evaluate if a newly recorded data point is significantly different to be deemed anomalous [88, 151].

But using an independent statistical classifier for each asset in the fleet is accompanied with distribution instabilities. Depending on the variance in asset data, distribution parameters would not be stable until certain amount of data describing the asset's normal operating regime is obtained. On the other hand, collective modelling of the fleet wide data is challenging as the assets operate over a wide range of environments, in various operating regimes, and can fail in multiple modes [90, 121]. Every asset has a unique behaviour and failure tendency, and ideally requires a classifier particularly suited for its operations [158].

The literature presents examples where the researchers have tried to address the issue of statistical instabilities. In the simplest form, similar assets are manually identified by the operators based on predetermined indicators, and an overall model is trained using the data from all units as a single dataset. This can be found in [203, 57, 97], where in every case the operators use a relevant parameter for for identifying the sub-fleets of similarly operating assets. Some researchers have also clustered the entire time series of condition monitoring data based on their Euclidean distances like in the case of [115, 101, 1]. In a comparatively complex collaborative approach, [123] modelled the functional behaviours of each unit using deep neural networks and identified the similar ones based on the amount of deviation in the neural network parameters. However, each of these applications are associated with their own set of constraints, which primarily are the lack of complete representation for the case of [203, 57, 97], dimensional complexity while evaluating the Euclidean distances in [115, 101, 1], and the necessary training data for each unit required to train the neural networks in the case of [123].

The closest application of collaborative learning for anomaly detection to that presented in Chapter 3 is found in [121]. [121] stresses the necessity of one class-classification for industrial systems owing to a wide range of possible operating regimes and rarity of failures. [121] also focus on early life monitoring where a given asset would not have sufficient data for training a robust classifier and propose that the asset rely on learning from other similar assets. However, their proposed solution relies on accumulating data from similar assets to a central location (or the target asset), and augmenting the features space to define a boundary for normal operation common to all similar assets. It must be noted that while the target problem is similar, [121] focus on feature alignment, whereas this thesis focuses on modelling an overall fleet behaviour and modifying it to suit individual assets. As such, the solution proposed in this thesis differs from the one presented in [121] in three aspects. First, the proposed hierarchical model is capable of identifying the asset clusters in the fleet, in contrast to [121] where it is assumed that all assets within the fleet are similar or known beforehand. Second, the operating regime targeted in this thesis is that of earlier durations compared to [121], where the assets they describe as new have 17,000 data points for 24-dimensional data.

Anomaly detection in asset operations has become increasingly important in the recent years due to widespread automation. Several researchers have shown that collaborative learning amongst the assets can help improve the performances of fault classification models, though with their own set of constraints. Anomaly detection is especially challenging during the early stages of asset operations where sufficient data are not available to model the corresponding regimes of operations.

2.1.4 Collaborative Prognosis

Collaborative prognosis is an emerging concept that aims at identifying clusters of similarly deteriorating assets and enabling learning across these clusters to improve the overall prognosis performance [158]. The earliest applications of collaborative prognosis involve identifying clusters of assets that operate in similar conditions and have encountered same failures, followed by sharing failure trajectories within these asset clusters. As a result, any given asset's data repository is enriched with failure trajectories imported from other assets. Prediction models are then trained using the enriched dataset, resulting in reduced error in the predictions [156]. It has been theoretically shown that collaborative prognostics is more cost-effective compared to self-learning (prognostics using the machine's own data [138]), and fleet-wide learning under the conditions when the individual assets do not have sufficient data for training the prognosis models [136].

The term *knowledge transfer* is used generally in the probabilistic machine learning literature to refer to methods that learn from multiple related datasets. In the context of fleet monitoring the majority of literature focusses on *transfer learning* where the domains share interpretable, parametrised models, and seek to improve predictions in a *target domain* given the information in a (more complete) *source domain* [126]. For example, [40, 46, 82] detect cracks over a number of domains by *fine-tuning* the parameters of a convolutional neural-network trained on a source domain to aid generalisation in the target.

Domain Adaptation is viewed as another variant of transfer learning in engineering applications (DA) [202, 109, 186]. These techniques define some mapping from domain data into a shared space (possibly one of the original domains) where a *single* model is used to make predictions. For example, [122] apply a neural network mapping for DA in the condition monitoring of a fleet of power plants. DA has also been investigated by (kernelised) linear projection, discussed in a structural health monitoring context by [51, 50] considering methods for knowledge transfer between simulated source and target structures, as well as a simulated source and experimental target structure [48]. Damage detectors have also been transferred between systems via DA in a group of tailplane structures using ground-test vibration data [21].

Identifying clusters of similar assets/ failures has also been the basis of many data-driven prognosis techniques apart from collaborative prognosis, where the initial investigations considered the quantification of *similarity* between the industrial systems [59], and tools for the *transfer* of data and/or models from *source* to *target* domains [122, 21, 47]. For example, [188] showed that in a system comprising of multiple assets and historical failures, prediction of a given asset is improved by identifying similar historical behaviours from a library of past failure data, and evaluating the best fit for the current failure's degradation curve. [43] used genetic algorithm to identify clusters of the most similar historical failure trajectories, which in turn improved the prediction accuracy of models corresponding to each of those identified clusters. Example implementation of this was shown for fatigue crack growth, drilling bit degradation, and degradation of a turnout system applications. [113] relied on collaborative learning to tackle the lack of sensing resources for the overall cohort of units, for the cases of both medical patients and industrial assets. Collaborative learning in this case was based on Markov models and selective sensing to address the problem of incomplete data per individual units. [24] discuss a case study with more than 30,000 machines for using clustering to identify a number of sets of assets with similar reliability behaviours. But unlike the examples listed above where a common prognosis model was used for the group of similar assets, collaborative prognosis aims at learning the deterioration across the similar asset groups and tailoring it for the individual assets.

2.1.5 Multi-Agent Frameworks for Collaborative Prognosis

The late 20th century saw rapid progress in telecommunications, metrology, and computing capabilities, propelling the realisation of Internet of Things, cloud computing, etc. [44, 163]. Embedded microprocessors in modern industrial assets enhance their digital capabilities, specifically by enabling local data analytics [120]. A fleet of modern assets therefore resembles a network of computers, enabling distributed deployment of the data-driven prognosis models at the asset level. Several distributed system frameworks and protocols have recently been postulated for the industrial systems [100, 9, 54]. One such distributed computing framework widely used in the industries, and especially for collaborative prognosis, is the framework of multi-agent systems (MAS) [73, 70].

The multi-agent systems framework originates from the field of Distributed Artificial Intelligence [55]. In an MAS, the overall system-goal is subdivided into agent-level goals depending on the knowledge and reasoning skills of the agents, which are the computing entities comprising the MAS. the level of intelligence and relationships among the agents can be defined by the user [17, 118, 125].

The use of MAS as a framework for decision-making/ control in manufacturing industries has been proposed by several researchers [183, 164], with the first ones being [41]. Using agents to represent parts, physical resources, and human operators [41] implemented a parts oriented scheduling approach. In another earlier paper, the authors demonstrate the use of Watchdog agent, which they have developed to prognose and diagnose the asset data based on the data flowing in from the sensors of the asset. Part agents and machine agents in [191] indulge in negotiation and cost evaluation to arrive at an optimal schedule. [105] use a supervisor agent which continuously monitors the degradation machine agents. The workload is assigned according to the machine's condition to optimise the maintenance cost. In a similar coordinator-based approach, [187] propose an MAS architecture where the agents make local decisions, but are controlled by a coordinator which monitors the overall system. The agents evolve by interacting with the coordinator via feedback loops. MAS are also used to implement various optimization heuristics to solve the NP-hard combinatorial problems of scheduling. [180] represented different departments, and machines with agents. These coordinated with one another to form an enterprise wide optimal maintenance plan using a memetic algorithm. While the distributed approach was not as economic as the centralised one, the time complexity (and thus scalability) was several orders of magnitude better. [193] implemented Ant Colony Optimisation on part-agents and machine-agents for task allocation and task sequencing. Several other use cases of collaborative agents for varied manufacturing industries applications can be found in the literature [68, 69, 85]. Distributed computing architectures such as the MAS have also been proposed for applications in supply chain management for example in [194]. Use of MAS for water supply systems can be found in [79, 71, 72]

Continuous learning in an MAS helps agents adapt to a real-life dynamic environment. The recent research towards collaborative prognosis focuses on distributed deployment of all constituting steps, ranging from identification of similar assets, training the models, and real time failure prediction using the MAS framework [17, 118, 125]. The concept of collaborative prognostics extends the concept of collaborative agents into the field of prognostics and health management. Collaborative agents share local information with each other in order to jointly achieve a given objective [171, 132]. [131] describe an analogy of such systems with the social organisation of humans as: each unit human has its nervous systems made up of the same physical components and operating laws, but individuals possess their own sophisticated and unique consciousness and behaviour. When implementing collaborative prognosis using an MAS framework, the assets (through their linked agents) behave like social entities, communicating with one another and making their own decisions for the prognosis applications [6]. It has been shown specifically for collaborative prognosis that the MAS framework enables more adaptable, scalable, resilient, flexible, and lean deployment than the former techniques which relied on centralised implementations [156]. Researchers have also worked towards evaluating the cost implications of the MAS architecture for collaborative prognosis, and in effect identifying the most suitable architecture for a given fleet of assets [137, 32].

This thesis specifically targets the scenario where the asset condition data is incrementally collected across a fleet of assets, and certain assets suffer due to sparse failure data, insufficient for training their corresponding prognosis models. Based on the discussion in this section, it is concluded that collaborative prognosis conceptually posits that a learning opportunity exits for the assets with sparse data to learn from similar other assets in the fleet. Moreover, sufficient infrastructural support exists for deploying the collaborative learning for the industrial fleets. Majority of the research in collaborative prognosis also focused on evaluating the suitable frameworks and architectures for deploying collaborative prognosis.

However, the literature does not present a solution to systematically achieve collaborative prognosis, as most existing approaches rely on the naive exchange of failure trajectories across the assets. The technique of statistical hierarchical modelling proposed in this thesis, and explained in Section 2.2, enables collaborative prognosis particularly for this purpose.

2.2 Statistical Hierarchical Modelling

This section introduces the reader to the concept of Statistical Hierarchical Modelling, and discusses related work on modelling data originating from a fleet of assets. It is proposed and demonstrated in this thesis that the technique of statistical hierarchical modelling potentially enables collaborative prognosis for an industrial assets fleet. The content of this section is in part taken or adapted from the co-authored paper [20].

2.2.1 Conceptual Introduction

Statistical hierarchical models provide solutions for those populations where the assumption of independence of the individuals does not hold true. As such, a single model cannot be justified for modelling the data. Common examples in a society can be found for example while modelling the grades of children in a city, where multiple schools exist. In this case a single model cannot be used to model the grades as the schools have an impact on a student's performance. And while the exact impact cannot be quantified, a hierarchical model enables modelling the data in the form it originates (in different clusters, which in this case are schools).

The intended application of the technique proposed in this thesis is for the assets with very sparse data, for example those recently in operation, or new operating conditions. In turn, model comparisons here are limited to parametric (or *shallow* [169]) methods of knowledge transfer, centred around interpretable models.

Multi-task learning (MTL) approach provides a generalised view of the statistical hierarchical modelling technique presented herewith. It assumes the predictors (*tasks*) are correlated over the fleet, and the parameters across the individuals are learnt commonly with equal importance. The combined inference enables *domain-specific* models to share information across related tasks, thus improving the accuracy for the individuals where data are limited [170].

Nevertheless, the examples of multi-task learning are less prevalent when modelling engineering assets. [184] successfully use a Gaussian process (GP) to learn correlations between tasks in a multi-output regression. The GP is built using a carefully specified kernel [13] to capture the task and inter-task relationships. Similarly, [110] apply correlated GPs to address the missing data problem over multiple sensors of a hydroelectric dam.

The results demonstrate successful knowledge transfer between measurement channels. Considering aerospace engines, [162] apply GPs for knowledge transfer between multiple axial measurement planes when interpolating temperature fields within an aircraft engine. Sharing information between planes significantly improves the spatial representation of the response.

Statistical Hierarchical Modelling offers a multi-task framework where a model is built with a 'hierarchy' of parameters, whereby domain-specific tasks are correlated via *shared* latent variables. Some recent, related applications include [36], who use hierarchical models to build corrosion models given evidence from multiple locations, and [140], where the results from a series of materials experiments (i.e. coupon samples) are combined to inform the estimation of material properties. Also, [30] implement hierarchical Gaussian mixture models to cluster simulated data that represent novelty detection for asset management; the model parameters are interpretable in terms of the data distribution, rather than the application domain.

2.2.2 Bayesian vs 'deep' knowledge transfer

As discussed in Section 2.1.4, popular applications of collaborative prognosis in the literature rely on deep learning techniques and are enabled by boosting the training dataset of the assets with fewer samples. While being able to improve the prognosis performance, such a solution does not provide for a formal technique to enable learning across the assets. In particular, the proposed technique of statistical hierarchical modelling compares with the deep learning as follows:

- Both address relative data sparsity across the assets, however the level of sparsity is method dependent. Deep learning methods are often suitable for complex features and big data whereas the hierarchical methods are suitable for standard measurements (e.g. vibration-based, power data, survival analysis) and interpretable models.
- Both improve predictions over asset clusters; however, the proposed hierarchical approach provides uncertainty quantification of the nested subgroups, enabling downstream (statistical) analyses, for example with experimental design or decision processes.
- Encoding domain (engineering) expertise while using statistical hierarchical models. For example, the knowledge that all turbines in a wind farm have the same maximum power, but the rate at which they limit to a maximum will depend on turbine location.

- Conversely, for neural networks, encoding domain expertise is difficult since they are nonparametric; in turn, the inferences (and model constraints) at different levels of fleet granularity are less intuitive.

A more general comparison of collaborative learning approaches for asset fleets is also provided in Table 2.1, to motivate the proposed method (labelled *Hierarchical Modelling, MTL*) and its advantages for asset fleets monitoring.

2.2.3 Brief Review of Federated Learning

A discussion on Federated Learning technique is provided herewith, which is a technique that targets a problem similar to that of collaborative prognosis but for mobile device applications.

Federated Learning (FL) technique originated for mobile device applications, primarily aiming at protecting user privacy. It is characterised by distributed training of a global predictive model, for applications such as text prediction or image classification. without the need for directly accessing the user data in their corresponding devices. In the pioneering paper from 2016, Federated learning was proposed by Google Inc to address the issue regarding use of private user data from their mobile devices to train common predictive models [93]. It was termed "Federated" because only a federation of network nodes, for example the mobile devices, participated in training the model at a given instance.

Over the years FL techniques have been deployed by major service providers and popularly proposed as an enabler of several data-sensitive applications such as text predictions or image classifications [12, 76, 65]. It has gained recent popularity across various applications also due to the increasing awareness about data privacy [196, 107]. FL today finds extended applications for those distributed systems where the local computations are orders of magnitudes faster than the communications across the nodes due to network size, or where data must not leave the nodes to protect the user privacies [93]. These constraints also apply to several industrial applications, including prognosis of asset fleets [154, 157].

The generic FL optimisation problem involves training a *single global statistical model* representing data stored across the nodes. This model is trained by optimising an objective function describing the overall system performance, which in turn involves jointly optimising local objective functions at the nodes [93]. The local objective functions might as well be different from the global objective function. The global objective function $F(w)$ for a network of m nodes is mathematically represented in (2.1) where the local objective functions are denoted by F_k for the k^{th} node, with w being their corresponding model parameters. p_k is weight associated with the k^{th} node. Choice of p_k varies across applications but popular choices are $p_k = \frac{n_k}{n}$ or $p_k = \frac{1}{m}$, where n_k is the amount of data at k^{th} node and n is the total data across the entire network.

$$\min_{w} F(w), \quad \text{where} \quad F(w) := \sum_{k=1}^{m} p_k F_k(w) \tag{2.1}$$

While (2.1) is the generic mathematical formulation, FL literature has instances where multiple objective functions have also been proposed [108].

The earliest adopting of FL beyond the field of mobile device communications was seen in medical sciences for the applications relying on the patient data to train the statistical models. Medical data is distributed across different locations such as hospitals, clinics, home-based devices, and even a person's smartphone. This includes personal data that cannot be shared across different locations. This poses a major challenge for the use of machine learning in healthcare, that can potentially be addressed using FL. The authors in [18] for example use FL to address the problem of sparse medical data. They use FL to train an SVM based binary classifier to predict hospitalisations during a target year for patients with heart diseases. The features used to train the algorithm were the patient history recorded in hospitals, or in their smartphones. In another paper, experiments have shown positive results when a Bayesian learning network was implemented to predict dyspnea in a distributed manner [86]. Here the data was distributed across five medical institutes and the model was circulated around each of the institutes during training, followed by the final model shared with all the institutions.

Majority of FL applications involve optimisations relying on Stochastic Gradient Descent as the basic training algorithm, which is popularly used for back-propagating the error in the SNNs [11, 201]. FedAvg is among the pioneering FL techniques for learning a single Artificial Neural Network (ANN) model for data distributed across network nodes. This is the underlying principle of FedAvg is that loss surfaces of sufficiently over-parameterised artificial ANNs are well behaved and escape bad local minima. Therefore, when two ANN models with the same parameter initialisations are trained independently on different subsets of IID data, naive averaging of their updated parameters can be used to obtain a single model describing combined data. Though primitive, Federated Averaging (FedAvg) is popular across domains like healthcare, mobile devices, home security systems, etc [107].

FedAvg has also been used for collaborative prognosis for sharing the failure information within the clusters of similar assets. For the case of asset prognosis, the training dataset comprises of the historical failure trajectories, which are distributed across several assets. Assets lie at the nodes of the network, that have varying instances of failure occurrences. The primary motivation for using FL for prognosis is that operators often do not want their asset data to be shared with their competitors [154, 165]. Exchanging failure trajectories to share information across the sub-fleets of similar assets also increases avoidable network communication costs [157]. In a series of papers the authors in [34, 35, 33] show that FedAvg is capable of training the recurrent neural network models for prognosis for using the failure trajectories form the sub-fleets of similarly deteriorating assets. However, the application of FedAvg was limited to the case where the assets only fail in a single failure mode. Since FedAvg requires data to be IID, each failure type has a prediction model specifically trained for its prediction. The performance of FedAvg steeply decreases with increasing diversity in the failure mode, and is strictly reliant on accurate identification of similarly deteriorating assets.

Nevertheless, FL is gaining increasing popularity in the industrial applications, with one of the main research focus being its ability to incorporate non-IID data for training the global model [146]. The problem of non-IID data is prevalent in prognosis applications and also in other general applications of FL [11, 106]. It should be noted that the motivation for using FL in the industries arises from the data security rather than collaborative learning [146, 130].

Table 2.1 A general comparison of collaborative learning approaches for asset fleets that highlights the advantages of statistical hierarchical modelling technique. This table is taken from [20].

Method	Single task Learning	Complete Pooling	Fine Tuning (Neural Nets)	Domain Adaptation	Neural Nets (MTL)	Hierarchical Modelling
Knowledge Transfer	none	data-level	pre-training on similar datasets	data-level (transformed)	correlated weights, shared layers	correlated or tied parameters
Interpretable	model dependent	model dependent	nonparametric/ black-box	model dependent	nonparametric/ black-box	yes
Task-specific or shared models	task-specific	shared only	task-specific (once retrained)	shared only	both	both (natural interpretation)
Dataset size	any (model dependent)	any (model dependent)	large	model dependent	large	any

2.3 Statistical Hierarchical Model for Asset Fleet data Modelling

This section provides an example of using a hierarchical linear regression model for fleet data, comprising of asset clusters and discusses various engineering applications of the statistical hierarchical modelling technique. The content of this section is taken from the co-authored paper [20] with minor adaptations.

2.3.1 Example using a Linear Regression Model for Asset Fleet Data

Consider an asset fleet with K clusters. The population data can then be denoted,

$$\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^K = \left\{ \{x_{ik}, y_{ik}\}_{i=1}^{N_k} \right\}_{k=1}^K \tag{2.2}$$

where $\mathbf{y}_k$ is target response vector for inputs $\mathbf{x}_k$ and $\{x_{ik}, y_{ik}\}$ are the i^{th} pair of observations in cluster k. There are N_k observations in each cluster and thus $\sum_{k=1}^K N_k$ observations in total. The aim is to learn a set of K parameters, one for each cluster, related to classification or regression tasks. Without loss of generality, this example focusses on the regression setting, where the tasks satisfy,

$$\{y_{ik} = f_k(x_{ik}) + \varepsilon_{ik}\}_{k=1}^K$$

i.e. the output is determined by evaluating one of K latent functions with additive noise ε_{ik}.

The mapping f_k is assumed to be correlated between asset clusters. In consequence, the models should be improved by jointly learning the parameters over the whole asset fleet. In machine learning where this is referred to as *multi-task learning*, hierarchical models enable modelling such data in statistical sense [94, 53].

In practice, while certain asset clusters might have rich, historical data, others (particularly those recently in operation) will have limited training data. In this setting, learning separate, independent models for each cluster will lead to unreliable predictions. On the other hand, a single regression of all the data (complete pooling) will result in poor generalisation. Instead, hierarchical models can be used to learn separate models for each cluster while encouraging task parameters to be correlated [126] – the established theory is summarised here.

Consider K linear regression models,

$$\left\{ \mathbf{y}_k = \Phi_k \alpha_k + \varepsilon_k \right\}_{k=1}^K \tag{2.3}$$

where $\Phi_k = [\mathbf{1}, \mathbf{x}_k]$ is the $N_k \times 2$ *design* matrix; α_k is the 2×1 vector of *weights*; and the noise vector is $N_k \times 1$ and normally distributed[2] $\varepsilon_k \sim \mathrm{N}\left(0, \sigma_k^2 \mathbf{I}\right)$. $\mathbf{1}$ is a vector of ones, $\mathbf{I}$ is the identity matrix, and $\mathrm{N}(m,s)$ is the normal distribution with mean m and (co)variance s. The likelihood of the target response vector is then,

$$
\begin{aligned}
\mathbf{y}_k | \mathbf{x}_k &\sim \mathrm{N}\left(\blacksquare_k \alpha_k,\ \sigma_k^2 \mathbf{I}\right) \\
\therefore \quad y_{ik} | x_{ik} &\sim \mathrm{N}\left(\alpha_1^{(k)} + \alpha_2^{(k)} x_{ik},\ \sigma_k^2\right)
\end{aligned}
\tag{2.4}
$$

In a Bayesian manner, one can set a shared hierarchy of prior distributions over the weights (slope and intercept) for the groups $k \in \{1, \dots, K\}$,

$$\{\alpha_k\}_{k=1}^K \overset{\text{i.i.d}}{\sim} \mathrm{N}\left(\mu_\alpha, \mathrm{diag}\left\{\sigma_\alpha^2\right\}\right) \tag{2.5}$$

$$\mu_\alpha \sim \mathrm{N}\left(\mathbf{m}_\alpha, \mathrm{diag}\left\{\mathbf{s}_\alpha\right\}\right) \tag{2.6}$$

$$\sigma_\alpha \overset{\text{i.i.d}}{\sim} \mathrm{IG}(a,b) \tag{2.7}$$

In words, (2.5) assumes that the weights $\{\alpha_k\}_{k=1}^K$ are normally distributed $\mathrm{N}(\cdot)$ with mean μ_α and covariance[3] $\mathrm{diag}\{\sigma_\alpha^2\}$. Similarly, (2.6) states that the prior expectation of the weights α_k is normally distributed with mean $\mathbf{m}_\alpha$ and covariance $\mathrm{diag}\{\mathbf{s}_\alpha\}$; (2.7) states that the prior deviation of the slope and intercept is inverse-Gamma distributed $\mathrm{IG}(\cdot)$ with shape and scale parameters a and b respectively.

Selecting appropriate prior distributions, and their associated hyperparameters $\{\mathbf{m}_\alpha, \mathbf{s}_\alpha, a, b\}$, is essential to the success of hierarchical models. In this work, prior elicitation is justified by encoding engineering knowledge in each case study as weakly informative priors [52]. The Directed Graphical Model (DGM) in 2.2 visualises the general hierarchical regression. The nodes show observed/latent variables as shaded/non-shaded respectively; arrows show conditional dependencies, and plates show multiple instances of sub-scripted nodes.

The K weight vectors α_k are correlated via the common latent variables $\{\mu_\alpha, \sigma_\alpha^2\}$. This does not restrict the covariance structure of the posterior distribution for $\{\alpha_k\}_{k=1}^K$ here since it is approximated using Markov Chain Monte Carlo (MCMC).

Via correlations in the posterior distribution, sparse clusters borrow statistical strength from those that are data-rich. Crucially, to *share* information between tasks, the parent nodes $\{\mu_\alpha, \sigma_\alpha^2\}$ must be inferred from the population data. In this way, the cluster parameters α_k are (indirectly) influenced by the wider population. Consider that, if $\{\mu_\alpha, \sigma_\alpha^2\}$ were

[2] In this first introductory example, the additive noise variance σ_k^2 is observed – in the next example, it is unobserved.

[3] The operator $\mathrm{diag}\{\mathbf{a}\}$ forms a square diagonal matrix with the elements from $\mathbf{a}$ on the main diagonal and zeros elsewhere.

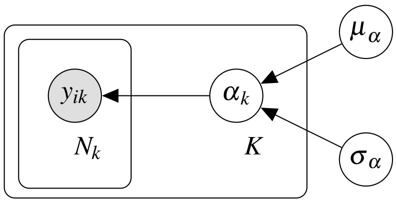


Fig. 2.2 DGM of hierarchical linear regression. This figure is taken from the co-authored paper [20].

fixed constants, rather than variables inferred from data, each model would be conditionally independent, preventing the *flow* of information between clusters [126].

2.3.2 Mixed-effects modelling

The hierarchical structure allows *effects* (i.e. interpretable latent variables) to be learnt at different levels, as well as 'prior' information. Specifically, the parameters of the model itself (2.3) can be learnt at the system, sub-fleet, or population level.

Returning to the regression example (2.3), consider that the variance σ_k^2 of the noise ε_k is in fact unknown. While one could learn K cluster-specific noise variance terms σ_k^2, it is typically assumed that the noise is equivalent across tasks. Sharing the parameter and inferring it from the population can significantly reduce the uncertainty in its prediction. Of course, this assumption should be justified given an understanding of the problem at hand; for example, the same sensing system collects all the asset condition data. In terms of notation, (2.3) remains the same, however, the domain-specific noise vector ε_k is now distributed $\varepsilon_k \sim \mathrm{N}\left(0, \sigma^2\mathbf{I}\right)$. The removal of subscript-k from the noise variance implies that the size of σ^2 remains the same while the number of the sub-fleets K increases (unlike α_k). Intuitively, σ^2 is now a *tied* parameter [126].

Similarly, it makes sense to also infer *effects* at the population level, to further reduce model uncertainty[4]. Throughout this work, it is assumed that *shared* effects also enter the model linearly, for the target response vector $\mathbf{y}_k$ and inputs $\mathbf{x}_k$,

$$\left\{\mathbf{y}_k = \underbrace{\Phi_k\alpha_k}_{\text{random}} + \underbrace{\Psi_k\beta}_{\text{fixed}} + \varepsilon_k\right\}_{k=1}^{K} \tag{2.8}$$

Where Ψ_k is some design matrix of inputs, and β is the corresponding vector of weights. Again, there is no subscript-k for β (like σ^2) as it is tied between sub-fleets. Following [94],

[4]For example, the intercept would be a shared parameter, with zero-mean, in a related linear regression of Hooke's law for several materials tests.

the β coefficients as considered *fixed effects*, as they are learnt at the population level and shared, while α_k are *random effects*, as they vary between *individuals*. Intuitively, a model with both fixed and random effects can be considered a *mixed* (effects) model [190, 52]. 2.3 shows the modified DGM of the hierarchical regression. The key differences are nodes outside of the K plate – these are the tied parameters, learnt at the population level.

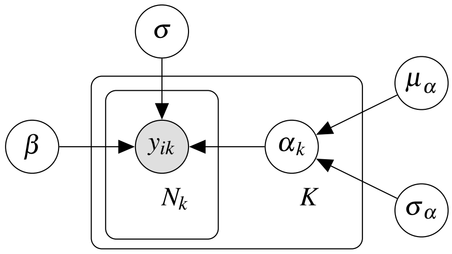


Fig. 2.3 DGM of hierarchical linear regression with mixed effects. This figure is taken from the co-authored paper [20].

One should also consider that interpreting mixed-effects models remains challenging, even when models are parametrised. If the effects are not (linearly) independent, the fixed and random coefficients can influence each other, making it difficult to reliably recover their relationships. In turn, the modelling assumptions must be carefully considered when emphasising interpretability.

2.3.3 Engineering Applications of Statistical Hierarchical Models

For the industries, statistical hierarchical models enable combined inference from the asset fleet data by learning a set of correlated models via shared higher level distributions. The shared higher level distributions ensure that the behaviour observed across other similar assets in the fleet is incorporated when the assets have sparse data. The parameters of the overall model are learnt using hierarchical Bayesian inference and provide robust variance reductions compared to the independent models inferred for the assets with sparse data [185, 52]. Hierarchical models therefore automatically incorporate collaborative learning across similar assets or sub-fleets, such that the assets with sparse data borrow statistical strength from those that are data-rich [126, 185]. Comprehensive information about statistical hierarchical modelling can be found in [52, 53].

The concept of knowledge transfer, from one machine to another, has often led to the development of population-based models [22, 59, 49] for prognosis applications in the industries [198]. One of the earliest applications use hierarchical models for inferring the Bernoulli parameters for reliability estimation of emergency diesel generators in separate nuclear power plants. They show that the hierarchical Bernoulli model was more accurate for modelling the collective “composite” and individual reliabilities of the generators, compared to the prevalent approach of analysing data from all generators as a single dataset. [36] use hierarchical models to build corrosion models given the data from multiple sources. An interesting application can also be found in [87] where hierarchical modelling was used for reliability estimation of new space crafts, which had only experienced few failures or in some cases no failures. Most other applications in the asset reliability estimations target modelling the times between failures. For example, used a hierarchical poisson process model to describe the times between failures of closing valves in the safety systems of nuclear plants. They used hierarchical modelling for median times between failures for a collection of valves experiencing different rates of failures over a period of observation. Similar other applications include [42], all commonly modelling the times between failures for various equipment. In structural health monitoring, hierarchical models find applications for learning multiple and correlated regression models for modal analysis [78, 77].

Of the more recent but fewer condition data-driven prognosis applications, demonstrated the benefits of hierarchical bayesian modelling for inferring the deterioration pattern of gas turbines operating in various conditions. Their model involved inferring the health index regression pattern of several gas turbines with respect to operating time, and was shown that hierarchical modelling is a statistically robust solution while learning the prediction function from data spanning across a large fleet of machines. [89] used hierarchical Bayesian neural networks for predicting the failure times of fatigue crack growth, where the focus was on quantifying the systemic heterogeneities across the assets rather than enhancing individual predictions.

It should be noted that it is critical to to establish an appropriate level of knowledge transfer between systems or domains while implementing multi-task learning for asset fleets. If information is inappropriately shared, this can lead to *negative transfer*, whereby population models prove worse than conventional (single task) learning. Importantly, the proposed model automatically determines an appropriate level of knowledge transfer, by learning the inter-task correlations from the data and combining this with engineering knowledge encoded as prior distributions within the hierarchical structure.

The resultant approach permits formal uncertainty quantification at various levels of the predictive model, and, in turn, various granularities of fleet behaviour (e.g. system-specific, condition-specific, or population-wide). Multiple levels of uncertainty quantification enable natural integration with decision processes, or experimental design procedures, considering the whole fleet. In turn, the model can be used to inform fleet interactions within a wider asset management programme.

The proposed technique of statistical hierarchical modelling makes inferences from observations at various levels – including larger groups and the aggregated population, apart from the cluster-level predictions. Inference of the joint population model (from task-specific observations) presents the knowledge transfer mechanism. The resultant structure produces both *shared* and *task-specific* models – this is not true for any of the benchmarks, which learn one of the two options (i.e. single-task learning, complete pooling, domain adaptation – 2.3.3).

2.4 Conclusions

This chapter presented a literature review of data-driven prognosis, collaborative prognosis, and anomaly detection in the industries to highlight the research gaps and identify a suitable technique to achieve collaborative learning for asset fleets. The technique of statistical hierarchical modelling is proposed to enable collaborative learning, after discussing and comparing various techniques from the literature such as deep learning. single-task learning, federated learning, etc. A brief discussion on engineering applications of statistical hierarchical modelling is also provided.

Based on the review presented in Section 2.1 it is concluded that data-driven prognosis is critical for modern industrial operations such as predictive maintenance. Primary business offerings of several companies now rely on data-driven prognosis at their foundation. However, a peculiar challenge exists for data-driven prognosis in the form of unavailability of failure data for certain assets. This is a challenge as the asset condition data is often incrementally obtained which requires a lead time before sufficient failure data can be collected, and particularly so for the rare failure modes or for unusual operating conditions. Failure-prediction models corresponding to the assets with sparse data are characterised by increased variance and unreliable predictions.

To that end, collaborative prognosis is an emerging concept that posits that the prognosis performance of the models associated with the assets with sparse data can be improved by identifying and learning from other similar assets in the fleet. The discussion in Section 2.1.4 explains that conceptual and infrastructural support exists for collaborative prognosis, but a technique to enable collaborative prognosis is a critical research gap. Similar problem also exists for anomaly detection in the asset condition data. The discussion in Section 2.1.3 explains how the problem of anomaly detection is interlinked and similar to the problem for data-driven prognosis of the asset fleets.

Section 2.2 highlights that statistical hierarchical models provide for a systematic technique modelling the failure data in asset fleets. Hierarchical models are characterised by multi-level modelling, such that the data is sampled from the independent models but the parameters of the independent models of the similarly operating assets in the fleet are sampled from the shared higher level distributions. Such a hierarchy ensures that the prognosis or anomaly detection models associated to the assets with sparse data can in fact rely on similar other data-rich assets to learn the failure behaviour.

The following chapter presents a hierarchical model for anomaly detection in the asset condition data.

Chapter 3

Statistical Hierarchical Model for Collaborative Anomaly Detection

It was highlighted in Chapter 1 and 2 that anomaly detection is critical for prognosis as the incipience of asset failures often manifest as anomalies in the time-series of condition data. Moreover, anomaly detection is also necessary to extract the failure trajectories corresponding to the historical failures. Historical failure trajectories are used as inputs while training prognosis models, adding to the criticality of anomaly detection.

Statistical classifiers are commonly used for anomaly detection in the time-series of asset condition data, which require training data corresponding to the asset's normal operations before acceptable accuracy and certainty can be achieved. But when deployed for the data originating from each asset independently, sufficient data corresponding to the normal operations is often not available in the early durations.

This chapter addresses research questions 1 and 2 and research objectives 2 and 3 outlined in Chapter 1 by presenting a solution to the above problem in the form of a hierarchical statistical model for anomaly detection. Analytical solutions to update the hierarchical model parameters are derived to enable anomaly detection in the assets with sparse data by identifying and learning from other similar assets. The hierarchical model presented herewith is capable of simultaneously identifying the sub-fleets of similarly behaving assets and enabling collaborative learning within these sub-fleets.

Results obtained with the hierarchical model show a significant improvement using the hierarchical model in terms of accuracy and variance for assets with sparse data, compared to independent modelling or a common fleet-wide model. Experiments are conducted to evaluate the classifier's performance for a range of proportion of the sparse data assets.

The continuing chapter is structured as: Section 3.1 presents the mathematical description of the independent and fleet-wide classifier models for anomaly detection, and also the description for extending the independent model to a hierarchical model. Asset fleet comprising of sub-fleets of similarly operating assets simulated for the experiments herewith and implementation of the hierarchical model for the simulated fleet are presented in Section 3.2. The same section also compares the performance of the hierarchical model with the case where the asset parameters were independently estimated. Section 3.3 summarises the key conclusions to this chapter.

3.1 Mathematical Description

This section introduces the reader to the independent and hierarchical models, in Sections 3.1.1 and 3.1.2 respectively, of the classifier used for anomaly detection.

Statistical classifiers are often recommended for anomaly detection in the time-series of asset condition data [88]. Statistical classifiers posit that the condition monitoring data generated during normal asset operations can be described using underlying distributions. Assuming that an asset commences operating in normal condition, the underlying density function $p(\theta)$, θ being its parameters, can be estimated to model that asset's normal operation data. Upcoming anomalies in asset operations cause a change in system dynamics, and induce deviation from its estimated density function. Statistical tests are used to evaluate if a newly recorded data point is significantly different to be deemed anomalous [88, 151].

The asset condition data are associated with intrinsic and extrinsic measurement errors caused by system instabilities and inefficiencies, even while the asset is operating in stable conditions. The combined random effect of error and fluctuations in the sensor measurements has been treated as multivariate Gaussian [14, 92, 160]. This chapter also considers a multivariate Gaussian for anomaly detection.

3.1.1 Independent Model for Anomaly Detection

Consider, a fleet comprising of I assets. Any given asset i is monitored using d sensors, measuring the internal and external parameters such as temperature, vibrations, pressure, etc. Each of which is a feature describing that asset's behaviour, and thus the n^{th} set of measurements from i^{th} asset can be represented as a vector $\mathbf{x}_{i,n} \in \mathbb{R}^d$.

If N_i measurements recorded from asset i over a given time period, then that asset's data can be represented as a vector $\mathbf{X}_i = [\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, ..., \mathbf{x}_{i,N_i}], \mathbf{X}_i \in \mathbb{R}^{d \times N_i}$.

Owing to the random nature of measurement noise, and assuming no manual interventions, the underlying distribution of an individual asset's data can be modelled using a multivariate

Gaussian $\mathbf{x}_{i,n} \sim \mathcal{N}(\mu_i, \mathbf{C}_i)$ where $\mu_i \in \mathbb{R}^d$ is the mean vector and $\mathbf{C}_i \in \mathbb{R}^{d \times d}$ is the covariance matrix.

$$p(\mathbf{x}_{i,n}|\mu_i, \mathbf{C}_i) = \frac{1}{\sqrt{(2\pi)^d |\mathbf{C}_i|}} \exp\Big(-\frac{1}{2}(\mathbf{x}_{i,n} - \mu_i)^T \mathbf{C}_i^{-1}(\mathbf{x}_{i,n} - \mu_i)\Big) \tag{3.1}$$

Maximum likelihood estimation can be used to evaluate $\hat{\mu}_i$ and $\hat{\mathbf{C}}_i$ values for $\mathbf{X}_i$. A graphical representation of an isolated independent asset model is shown in Figure 3.1. The following section describes extending the independent asset model to a hierarchical model.

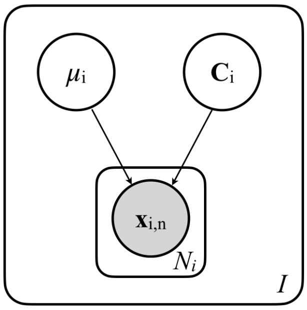


Fig. 3.1 Graphical representation of modelling an asset's data as multivariate Gaussian

3.1.2 Hierarchical Model for Anomaly Detection

A fleet often comprises of assets which are similar by their operational behaviour. This could be because certain assets have the same base model, or they may be operating in similar conditions [83, 102]. It gives rise to the presence of statistically homogenous *asset clusters* within the fleet. The challenges related to distribution instabilities introduced in Sections 1.1.3 and 2.1.3 can be alleviated if the individuals comprising such a cluster are jointly modelled with a common overlying distribution of their individual distribution parameters.

Hierarchical model of the asset fleet presented here mathematically formulates this idea by defining distributions at two levels. The parameters describing the distributions of individual asset data are considered to be sampled from their corresponding higher level distributions. The higher level distributions are shared by the asset clusters, and therefore jointly resemble the operating regimes of the assets comprising those clusters. The higher level distributions are chosen as the conjugate priors of the asset level distribution parameters. Estimated asset level parameters are weighed more towards the higher level distribution when the asset does not possess sufficient data. However, as more data is accumulated over time, the weight shifts towards the asset's own data and eventually becomes equivalent to an independent model. This enables an asset with insufficient data in its early phase of operations to collaboratively learn from similar other assets containing more data.

For the case of asset fleets, Normal-Inverse Wishart are chosen as the higher level distributions. These are the natural conjugate priors for a multivariate Gaussian with unknown mean and covariance. Concretely, the parameters $(\mu_i, \mathbf{C}_i)$ describing i^{th} asset are believed to be sampled from higher distributions as $\mu_i \sim \mathcal{N}(\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)$ and $\mathbf{C}_i \sim \mathcal{IW}(\Lambda_k, \alpha_k)$ where $k = 1, 2, ..., K$ represents the cluster index and $(\mathbf{m}_k \in \mathbb{R}^d, \beta_k \in \mathbb{R}, \Lambda_k \in \mathbb{R}^{d\times d}, \alpha_k \in \mathbb{R})$ are the parameters of cluster level distributions.

$$p(\mu_i|\mathbf{m}_k, \beta_k, \mathbf{C}_i) = \mathcal{N}(\mu_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i) = \sqrt{\frac{\beta_k^d}{(2\pi)^d|\mathbf{C}_i|}} \exp\Big(-\frac{\beta_k}{2}(\mu_i - \mathbf{m}_k)^T \mathbf{C}_i^{-1}(\mu_i - \mathbf{m}_k)\Big) \tag{3.2}$$

$$p(\mathbf{C}_i|\Lambda_k, \alpha_k) = \mathcal{IW}(\mathbf{C}_i|\Lambda_k, \alpha_k) = \frac{|\Lambda_k|^{\alpha_k/2}}{2^{\alpha_k d/2}\Gamma_d(\frac{\alpha_k}{2})}|\mathbf{C}_i|^{-(\alpha_k+d+1)/2} \exp\Big(-\frac{1}{2}Tr(\Lambda_i \mathbf{C}_i^{-1})\Big) \tag{3.3}$$

where Γ is the multivariate Gamma function, and $Tr()$ is the trace function.

As it can be observed that, at higher level lies a mixture of Normal-Inverse Wishart distributions from which pairs of $(\mu_i, \mathbf{C}_i)$ are sampled. The probability density function for a given $(\mu_i, \mathbf{C}_i)$ pair conditional on higher level parameters therefore can therefore be written as:

$$p(\mu_i, \mathbf{C}_i|\mathbf{m}_k, \beta_k, \Lambda_k, \alpha_k) = \sum_{k=1}^{K} \big[\pi_k \mathcal{N}(\mu_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\Lambda_k, \alpha_k)\big] \tag{3.4}$$

Where $\pi_k \in \mathbb{R}$ and $\sum_{k=1}^{K} \pi_k = 1$ is the proportion of assets belonging to k^{th} cluster. Individual asset data are further sampled from this $(\mu_i, \mathbf{C}_i)$ pair.

Therefore, probability density function for complete data for an asset i is:

$$p(\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, ..., \mathbf{x}_{1,N_i}) = \prod_{n=1}^{N_i} \Big[\mathcal{N}(\mu_i, \mathbf{C}_i) \sum_{k=1}^{K} \big[\pi_k \mathcal{N}(\mu_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\Lambda_k, \alpha_k)\big]\Big] \tag{3.5}$$

probability density function of the entire fleet data across all assets (represented by $\mathbf{X}$) is:

$$p(\mathbf{X}) = \prod_{i=1}^{I} \Bigg[\prod_{n=1}^{N_i} \Big[\mathcal{N}(\mu_i, \mathbf{C}_i) \sum_{k=1}^{K} \big[\pi_k \mathcal{N}(\mu_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\Lambda_k, \alpha_k)\big]\Big]\Bigg] \tag{3.6}$$

For a given set of $(\mu_i, \mathbf{C}_i, \mathbf{m}_k, \alpha_k,)$, the above function is also the likelihood of the data. Obtaining estimates of $(\mu_i, \mathbf{C}_i, \mathbf{m}_k, \alpha_k,)$ parameters would therefore require maximising the log of above probability function with respect to the parameters. The required log-likelihood objective function of the entire dataset for given parameter values is:

$$\log(p(\mathbf{X})) = \sum_{i=1}^{I}\sum_{n=1}^{N_i} \log(\mathcal{N}(\mu_i, \mathbf{C}_i)) + \sum_{i=1}^{I} \log\Big(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\Lambda_k, \alpha_k)\Big) \tag{3.7}$$

However, it can be observed that, due to presence of summation $\sum_{k=1}^{K}$ within log() function in the second term, analytically evaluating partial derivatives and equating them to zero is not straightforward, because both LHS and RHS of the final equations would comprise of unknown parameters. The next section explains an iterative expectation maximisation (EM) algorithm that solves this problem.

Model Parameters Estimation

Maximising the log-likelihood in (3.7) is difficult specifically because the clusters within the fleet and their constituent assets are not predetermined. The data is therefore in a sense incomplete.

A latent (hidden) binary variable matrix $\mathbf{z} \in \{0,1\}^{I\times K}$ is introduced to complete the data, such that $\mathbf{z}_{i,k} = 1$ if the i^{th} asset belongs to the k^{th} cluster. For a given asset i and set of distribution parameters, the probability of $\mathbf{z}_{i,k} = 1$ is therefore given by:

$$p(\mathbf{z}_{i,k}|\theta) = \pi_k \tag{3.8}$$

This, if evaluated across all values of k, and $\mathbf{z}_i^{th}$ vector of $\mathbf{z}$ would be:

$$p(\mathbf{z}_i|\theta) = \prod_{k=1}^{K} [\pi_k]^{\mathbf{z}_{i,k}} \tag{3.9}$$

Where θ represents the set of parameters $(\mathbf{m}_k, \beta_k, \Lambda_k, \alpha_k, \pi_k)$.

Moreover, The probability of $(\mu_i, \mathbf{C}_i)$ conditioned on $\mathbf{z}_{i,k} = 1$ is:

$$p(\mu_i, \mathbf{C}_i|\mathbf{z}_{i,k} = 1, \theta) = \mathcal{N}(\mu_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\Lambda_k, \alpha_k) \tag{3.10}$$

This, again if evaluated across all values of k is given by:

$$p(\mu_i, \mathbf{C}_i | \mathbf{z}_i = 1, \theta) = \prod_{k=1}^{K} \left[\mathcal{N}(\mu_i | \mathbf{m}_k, \beta_k^{-1} \mathbf{C}_i) \mathcal{IW}(\mathbf{C}_i | \Lambda_k, \alpha_k) \right]^{\mathbf{z}_{i,k}} \tag{3.11}$$

Probability of $(\mu_i, \mathbf{C}_i, \mathbf{z}_i)$ can therefore be evaluated simply by multiplying (3.9) and (3.11) as:

$$p(\mu_i, \mathbf{C}_i, \mathbf{z}_i | \theta) = \prod_{k=1}^{K} \left[\pi_k \mathcal{N}(\mu_i | \mathbf{m}_k, \beta_k^{-1} \mathbf{C}_i) \mathcal{IW}(\mathbf{C}_i | \Lambda_k, \alpha_k) \right]^{\mathbf{z}_{i,k}} \tag{3.12}$$

Continuing similar to (3.5) and (3.6), the complete data probability for a given set of parameters θ is given by:

$$p(\mathbf{X}, \mathbf{z} | \theta) = \prod_{i=1}^{I} \left[\prod_{n=1}^{N_i} \left[\mathcal{N}(\mathbf{x}_i | \mu_i, \mathbf{C}_i) \prod_{k=1}^{K} \left[\pi_k \mathcal{N}(\mu_i | \mathbf{m}_k, \beta_k^{-1} \mathbf{C}_i) \mathcal{IW}(\mathbf{C}_i | \Lambda_k, \alpha_k) \right]^{\mathbf{z}_{i,k}} \right] \right] \tag{3.13}$$

The graphical representation shown in Figure 3.2 describes the hierarchical modelling for whole fleet data, including the hidden cluster indicator $\mathbf{z}$.

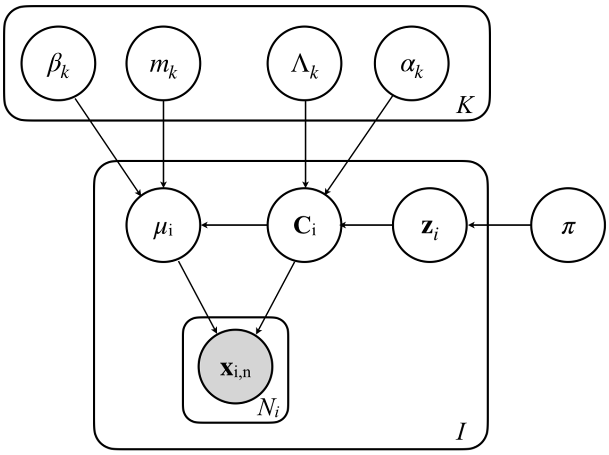


Fig. 3.2 Graphical representation of hierarchically modelled fleet data. Individual asset data are modelled as multivariate Gaussians, whose mean and covariance parameters are sampled from higher level Normal-Inverse Wishart distributions respectively

The complete data log-likelihood for a given set of parameters θ thus equates to:

$$\log(p(\mathbf{X}, \mathbf{z} | \theta)) = \sum_{i=1}^{I} \sum_{n=1}^{N_i} \log(\mathcal{N}(\mathbf{x}_i | \mu_i, \mathbf{C}_i)) + \sum_{i=1}^{I} \sum_{k=1}^{K} \mathbf{z}_{i,k} \log \left(\pi_k \mathcal{N}(\mu_i | \mathbf{m}_k, \beta_k^{-1} \mathbf{C}_i) \mathcal{IW}(\mathbf{C}_i | \Lambda_k, \alpha_k) \right) \tag{3.14}$$

To maximise the complete data log-likelihood function in (3.14), (3.14) must be differentiated with respect to individual parameters to obtain the corresponding maxima. However, the values of $\mathbf{z}_{i,k}$ are unknown, and therefore the partial derivative equations are not solvable.

The Expectation Maximisation (EM) algorithm addresses this problem of parameter estimation via looped iterations through two steps: the Expectation(E)-step, and the Maximisation(M)-step which are explained in the following subsections. Here again, θ are the model parameters and the parameters corresponding to t^{th} iteration are written as θ^t.

In the E-step, a function $Q(\theta,\theta^t)$ is computed which is the expectation of the complete data log-likelihood w.r.t. the distribution of hidden variable $\mathbf{z}$ conditioned over the incomplete data $\mathbf{X}$ and θ^t parameter values. Concretely,

$$Q(\theta,\theta^t) = E_{z|X,\theta^{t-1}}\{\log(l(\mathbf{X},\mathbf{z}|\theta))\} \tag{3.15}$$

Therefore the $\mathbf{z}$ terms are replaced by their expected values for the given incomplete data $\mathbf{X}$ and θ^t parameter values, and the other terms in $Q(\theta,\theta^t)$ depend on θ.

In the M-step, the values of parameters for the next $(t+1)^{th}$ iteration θ^{t+1} of the E-step are evaluated by maximising $Q(\theta,\theta^t)$ over θ, but treating $\mathbf{z}$ terms as constants.

$$\theta^{t+1} = \arg\max_{\theta} Q(\theta,\theta^t) \tag{3.16}$$

Estimated values of model parameters at M-step of every EM iteration are presented in (3.17) to (3.22), where the "$\gamma_{i,k}$" terms are the expected $\mathbf{z}_{i,k}$ values from the previous E-step. The estimates for α_k at M-steps can be obtained using any non-linear optimisation routine. Derivations of the E- and M- steps for our application are shown in Appendix A.

$$\frac{1}{\hat{\beta}_k} = \frac{\sum_{i=1}^{I}\gamma_{i,k}(\mu_i-\mathbf{m}_k)^T\mathbf{C}_i^{-1}(\mu_i-\mathbf{m}_k)}{d\sum_{i=1}^{I}\gamma_{i,k}} \tag{3.17}$$

$$\hat{\mathbf{m}}_k = \left[\sum_{i=1}^{I}\gamma_{i,k}\mathbf{C}_i^{-1}\right]^{-1}\left[\sum_{i=1}^{I}\gamma_{i,k}\mathbf{C}_i^{-1}\mu_i\right] \tag{3.18}$$

$$\hat{\Lambda}_k = \left[\alpha_k\sum_{i=1}^{I}\gamma_{i,k}\right]\left[\sum_{i=1}^{I}\gamma_{i,k}\mathbf{C}_i^{-1}\right]^{-1} \tag{3.19}$$

$$\hat{\pi}_k = \frac{\sum_{i=1}^{I}\gamma_{i,k}}{I} \tag{3.20}$$

$$\hat{\mu}_i = \frac{1}{N_i+\sum_{k=1}^{K}\beta_k\gamma_{i,k}}\left[\sum_{n=1}^{N_i}\mathbf{x}_{i,n}+\sum_{k=1}^{K}\beta_k\gamma_{i,k}\mathbf{m}_k\right] \tag{3.21}$$

$$\hat{\mathbf{C}}_i = \frac{\sum_{n=1}^{N_i}(\mathbf{x}_{i,n}-\mu_i)(\mathbf{x}_{i,n}-\mu_i)^T + \sum_{k=1}^{K}\beta_k\gamma_{i,k}(\mu_i-\mathbf{m}_k)(\mu_i-\mathbf{m}_k)^T + \sum_{k=1}^{K}\gamma_{i,k}\Lambda_k}{N_i + \sum_{k=1}^{K}\gamma_{i,k}\alpha_k + d + 2} \tag{3.22}$$

Parameters for the zeroth iteration are randomly initialised, and the estimates are believed to have converged when their evaluated values are consistent over consecutive iterations or when the complete data log likelihood in (3.14) ceases to increase any further with more iterations.

The initialisation of parameters can also vary by application. Generally it was observed here that, the asset level parameters (i.e. $(\mu_i, \mathbf{C}_i)\ \forall\ i \in \{I\}$) were best initialised by the standard maximum log-likelihood estimator for the asset's Gaussian model. While initialising the higher level parameters, β_k were best initialised at low values and α_k as equal to the dimension of the data. These ensured wider search space in the early iterations. $(\mathbf{m}_k, \Lambda_k)\ \forall\ k \in \{K\}$ initialised randomly around the observed data values, but ensuring that the initial Λ_k were positive definite matrices.

The steps followed for hierarchical model parameters estimation, including the initialisation in the experiments described here and EM iterations, are summarised in Algorithm 1. In Algorithm 1, $E(x_{i,n})$ represents the expectation of $x_{i,n}$ vector, $rand(d)$ and $rand(d,d)$ functions generate random real numbered matrices of (d) and $(d \times d)$ dimensions respectively, and $p(clust_i = k)$ represents the overall data likelihood for the i^{th} asset, assuming that the i^{th} asset belongs to the cluster k. Moreover, the terms on the RHS in the M-step are the values from the previous iterations, except $\gamma_{i,k}$ which are evaluated at the corresponding E-step.

Algorithm 1: Estimating the parameters for K sub-fleets and d dimensional data

Result: Estimated hierarchical model parameters

for *each asset i* **do**

  $\mu_i \leftarrow \frac{\sum_{n=1}^{N_i} x_{i,n}}{N_i}$;

  $C_i^{(n,m)} \leftarrow E\big((x_{i,n} - E(x_{i,n}))(x_{i,m} - E(x_{i,m})\big)$;

end

for *each cluster k* **do**

  $\beta_k \leftarrow 0.001$;

  $\alpha_k \leftarrow d$;

  $(\mathbf{m}_k, \Lambda_k) \leftarrow \big(rand(d), rand(d \times d)\big)$;

end

while *Iter < 20* **do**

  The E-step:

  for *each asset i and cluster k* **do**

    $\gamma_{i,k} \leftarrow \frac{p(clust_i=k)}{p(clust_i=1)+p(clust_i=2)+...+p(clust_i=k)}$;

  end

  The M-step:

  for *each asset i* **do**

    $\hat{\mu}_i \leftarrow \frac{1}{N_i+\sum_{k=1}^{K}\beta_k\gamma_{i,k}}\left[\sum_{n=1}^{N_i}\mathbf{x}_{i,n} + \sum_{k=1}^{K}\beta_k\gamma_{i,k}\mathbf{m}_k\right]$;

    $\hat{\mathbf{C}}_i \leftarrow \frac{\sum_{n=1}^{N_i}(\mathbf{x}_{i,n}-\mu_i)(\mathbf{x}_{i,n}-\mu_i)^T + \sum_{k=1}^{K}\beta_k\gamma_{i,k}(\mu_i-\mathbf{m}_k)(\mu_i-\mathbf{m}_k)^T + \sum_{k=1}^{K}\gamma_{i,k}\Lambda_k}{N_i+\sum_{k=1}^{K}\gamma_{i,k}\alpha_k+d+2}$;

  end

  for *each cluster k* **do**

    $\frac{1}{\hat{\beta}_k} \leftarrow \frac{\sum_{i=1}^{I}\gamma_{i,k}(\mu_i-\mathbf{m}_k)^T\mathbf{C}_i^{-1}(\mu_i-\mathbf{m}_k)}{d\sum_{i=1}^{I}\gamma_{i,k}}$;

    $\hat{\mathbf{m}}_k \leftarrow \left[\sum_{i=1}^{I}\gamma_{i,k}\mathbf{C}_i^{-1}\right]^{-1}\left[\sum_{i=1}^{I}\gamma_{i,k}\mathbf{C}_i^{-1}\mu_i\right]$;

    $\hat{\Lambda}_k \leftarrow \left[\alpha_k\sum_{i=1}^{I}\gamma_{i,k}\right]\left[\sum_{i=1}^{I}\gamma_{i,k}\mathbf{C}_i^{-1}\right]^{-1}$;

    $\hat{\pi}_k \leftarrow \frac{\sum_{i=1}^{I}\gamma_{i,k}}{I}$;

    $\alpha_k \leftarrow \mathrm{BFGS}_{max}\Bigg(\frac{1}{2}\alpha_k\log|\Lambda_k|\sum_i\gamma_{ik} - \frac{d}{2}\log(2)\alpha_k\sum_i\gamma_{ik} - \log\left(\Gamma_d\left(\frac{\alpha_k}{2}\right)\right)\sum_i\gamma_{ik} - \frac{1}{2}(\alpha_k+d+1)\sum_i\gamma_{ik}\log|C_i|\Bigg)$;

  end

  $Iter \leftarrow Iter + 1$;

end

return: $(\mu_i, \mathbf{C}_i, \beta_k, \alpha_k, \Lambda_k, \mathbf{m}_k)\ \forall\ i,k \in I,K$ respectively.

3.2 Example Implementation

This section discusses the experiments conducted to demonstrate and evaluate the performance of the hierarchical model for anomaly detection. Performance of the hierarchical model is also compared with independent and fleet-wide modelling of the assets.

Independent modelling does not consider the presence of similar assets in the fleet. Therefore, the $(\hat{\mu}_i, \hat{C}_i)$ estimates for every asset, obtained via independent modelling, correspond to their maximum likelihood estimates based on that asset's data only. These estimates are evaluated according to (3.23) and (3.24).

$$\hat{\mu}_i = \frac{\sum_{n=1}^{N_i} x_{i,n}}{N_i} \tag{3.23}$$

$$\hat{C}_i^{(n,m)} = E\big((x_{i,n} - E(x_{i,n}))(x_{i,m} - E(x_{i,m})\big) \tag{3.24}$$

Where $\hat{C}_i^{(n,m)}$ represents the $(n,m)^{th}$ entry of the estimated covariance matrix $\hat{C}_i$, and $E(x_{i,n})$ represents the expectation of $x_{i,n}$ data vector.

Experimental cases, and the performance metric used for evaluating and comparing both modelling approaches are described in the following subsections. Section 3.2.1 explains the synthetic dataset used for the experiments, Section 3.2.3 describes the evaluation metric, and finally Sections 3.2.2 and 3.2.4 presents the experimental results to compare the performances of hierarchical and independent modelling techniques.

3.2.1 Experimental Data

Synthetic datasets representing a fleet of assets, containing sub-populations of similar assets, were used for the experiments. These constituted the *training* and the *testing* datasets.

Training dataset

The data generation method described here ensured that the fleet comprised of coherent sub-populations of assets, and also that no two assets in the fleet were identical.

The training dataset comprised of multidimensional samples of assets' condition data over a period of their normal operation and collected across the entire fleet. The condition data for each asset comprised of points randomly sampled from a Gaussian distribution, with constant mean and covariance. This ensured that the simulated asset data was equivalent to a real asset operating in steady condition but with associated noise and fluctuations. The means of the underlying Gaussians were considered to be the equivalents of the asset model types, and the covariances of the Gaussians were considered to be the equivalents of their operating conditions.

Different asset model types are designed to operate in different ranges. Therefore, the assets belonging to the same model type are expected to operate within a certain permissible range. This was represented in the training dataset by defining ranges for the Gaussian means of assets belonging to separate model types. Similarly, the operating condition of an asset determines how much variation is caused in its condition data. For example, older engines are expected to have higher vibrations than the newer ones, and therefore induce larger variation from their mean vibrations value. This was represented in the dataset by defining a set of possible covariance matrices that an asset's Gaussian can be associated with.

Before simulating the assets, separate ranges for each feature were defined. Each set of ranges represented a separate model type present in the fleet. Moreover, a set of covariance matrices was also defined. While simulating an asset, its model type and operating condition were first characterised. Following which, the multidimensional mean of that asset's underlying Gaussian distribution was randomly selected within the range of its corresponding model type. Similarly, the covariance matrix corresponding to the asset's operating condition was selected from the predefined set of covariances. From this Gaussian, number of points were sampled, which represented that asset's condition data collected over a period of its normal operation. The same process was repeated for all assets comprising the fleet, and the final collection of points for assets constituted the training dataset.

Testing dataset

The testing dataset for any given simulated asset described in Section 3.2.1 was a mixture of points sampled from that asset's true underlying distribution and points sampled from an anomalous distribution. The anomalous distribution was generated by inducing systematic deviation from the true underlying distribution. This deviation was induced in the form of change in the mean and covariance of the true distribution. A large number of points were sampled from both true and anomalous distribution to ensure good statistics.

Consider a given asset i in the fleet, whose true underlying distribution had the mean and covariance values μ_i and C_i respectively. The anomalous distribution for this asset would be a multivariate Gaussian of the same dimension, but with its underlying mean and covariance being $\mu_i + l$ and $L. * C_i$ where, l and L are the deviations induced into the true mean and covariance values. The induced deviations were constant across all assets. Moreover, both l and L were varied across a wide range to study the sensitivity of the classifiers with respect to the Gaussian's mean and covariance.

A schematic description of how the normal and anomalous data for the simulated assets were generated is shown in Figure 3.3. This figure shows an example of generating normal and anomalous data for a two dimensional data set, where the regions defined for separate model types are shaded in colour and the set of covariances are shown using ellipses. And while the procedure is the same for five dimensional data, the regions in space representing the model types have been widened in Figure 3.3 for easier representation.

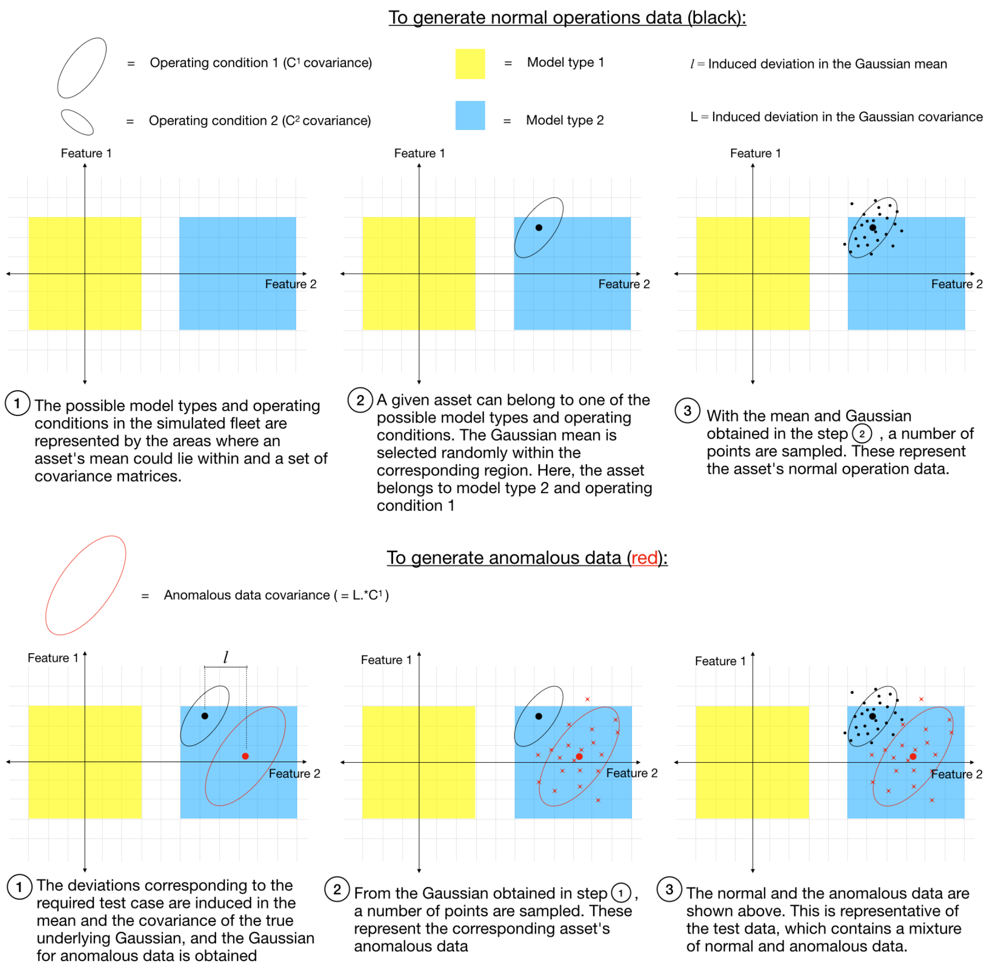


Fig. 3.3 A schematic representation describing how the normal and anomalous data were generated for the experiments. The procedure is shown here for a two dimensional dataset as an example.

Experimental specifications

The simulated fleet used for the experiments discussed here comprised of 800 assets. The assets could each belong to either of the two possible operating conditions and to either of the two possible model types. Therefore, the fleet comprised of total four clusters of assets, represented by each combination of the operating condition and the model type. All clusters contained the same number of assets (i.e. 200 assets per cluster).

The simulated condition data was five dimensional. All asset means for those belonging to the first model type lay within the range $(-25, 25)$, and for the second model type lay within the range $(275, 325)$. Similarly, the two covariance matrices corresponding to the operating conditions are shown in (3.25) and (3.26). The ranges for means and the two covariance matrices were arbitrarily chosen.

$$C^1 = \begin{bmatrix} 16.68 & 5.43 & 3.28 & -2.31 & 1.76 \\ 5.43 & 22.05 & -3.74 & -1.11 & -1.14 \\ 3.28 & -3.74 & 18.72 & 3.91 & -3.19 \\ -2.31 & -1.11 & 3.91 & 20.87 & 4.00 \\ 1.76 & -1.14 & -3.19 & 4.00 & 23.12 \end{bmatrix} \tag{3.25}$$

$$C^2 = \begin{bmatrix} 55.59 & 3.39 & 3.24 & -2.00 & -3.95 \\ 3.39 & 55.75 & 1.22 & -24.02 & -3.76 \\ 3.24 & 1.22 & 55.83 & 15.29 & 1.78 \\ -2.00 & -24.02 & 15.29 & 63.69 & 11.21 \\ -3.95 & -3.76 & 1.78 & 11.21 & 23.12 \end{bmatrix} \tag{3.26}$$

Where the superscript represents the cluster id. Moreover, the assets comprising the fleet held different amount of data (number of points sampled from its underlying Gaussian). Each asset could have either low, medium, or high amount of data. Assets belonging to the low data category held only 5 data points. Assets belonging to the medium and high data category contained 20 and 100 data points respectively. To make the setup clear, the corresponding values of the variables defined and derived in Section 3.1 are summarised in Table 1.

Table 3.1 The values of various parameters introduced in Section 3.1.

Parameter	**Value**
I	800
d	5
K	4
N_i	5 (for low data category); 20 (for medium data category); 100 (for high data category)

As an example, consider an asset belonging to the first model type and first operating condition. Let this asset belong to the "medium" data category. To simulate this asset, its mean was first selected as a random point with features lying within the range $(-25, 25)$. This mean was $(10.05, -15.95, 4.94, -4.24, 0.68)$. Next, with this mean and C^1 from (3.25 and 3.26) as the covariance, 20 points were randomly sampled. 20 points were sampled because this asset belonged to the medium data category. An example of the condition data for this asset is shown in Table 2. The remaining 799 assets in the fleet were similarly simulated based on their model type, operating condition, and the category they belonged to. The complete training and testing datasets can be found at: https://github.com/Dhada27/Hierarchical-Modelling-Asset-Fleets

Table 3.2 An example of condition data for a medium data category asset.

Measurement number	x_1	x_2	x_3	x_4	x_5
1	17.33	-23.02	1.88	-3.38	6.06
2	12.29	-14.77	2.87	-0.40	-2.80
3	9.93	-15.19	6.12	2.69	-2.52
...	...	...	...	...	...
19	8.28	-16.18	4.05	-0.21	-2.76
20	11.39	-13.20	12.56	-8.65	-1.26

The proportion of assets belonging to the low data category were varied across a wide range from 0.1 to 0.9. The remaining assets were evenly divided into medium and high data categories. For example, if 0.3 proportion of assets belonged to the low data category, then 0.35 proportion of assets belonged to high and medium data category each. Moreover, all clusters contained the same number of assets belonging to either of the three categories. Given this dataset, the goal for an anomaly detection algorithm was to model the assets' normal operation by estimating the parameters of the underlying Gaussians. There was no indicator for the algorithm to know which cluster a given asset belonged to.

The testing dataset for each asset comprised of 1500 points randomly sampled from the true underlying distribution, and 1500 points sampled from the anomalous distribution. The deviations l and L for the anomalous distributions were each varied while keeping the other constant, so that the sensitivity of the algorithms with respect to either parameters could be studied. Values of l were varied across $\{0, 5, 10, 20, 50, 100\}$ while keeping L fixed at 1, and the values of L were varied across $\{1, 1.5, 2, 5, 10\}$ while keeping l fixed at 0.

3.2.2 Experimental Design

The experiments conducted to study the hierarchical model for anomaly detection were aimed at analysing the performance of the hierarchical model, and also comparing its performance with the independent and fleet wide models to address research objectives 2 and 3 outlined in Chapter 1 for collaborative anomaly detection. As such, the experiments involved comparing four learning scenarios as explained below

1. **Independent Learning** In the first scenario, the assets were capable of learning from their own data only. This means that the only source of information for estimating the parameters of the underlying Gaussian was the given asset's condition data only. The mean and covariance estimates in this scenario were evaluated according to the standard maximum likelihood estimation in (3.23) and (3.24).

2. **Learning from similar assets** In this scenario, the hierarchical model for the fleet was implemented. Clusters of similar assets were identified, and the parameters for the hierarchical model were estimated using the EM algorithm as explained in Section 3.1. The EM steps were iterated 20 times, and the values of $\hat{\mu}_i$ and $\hat{C}_i$ after the 20^{th} iteration were treated as the final estimates of hierarchical modelling. 20 iterations were deemed sufficient for parameter estimation because the overall data log likelihood did not increase any further. The value of K, which are the number of clusters present in the fleet was set to its true value 4.

3. **Learning from all** The third scenario was similar to the one in case 2 above, but with the difference being in this scenario the assets did not have a sense of identifying similar assets. This means that a given asset here learnt from all other assets in the fleet. To model this scenario, the same steps as those in case 2 were followed, but the value of K was set to 1. As a result, the entire fleet was treated as one cluster and the density function parameters of all assets shared a common underlying distribution.

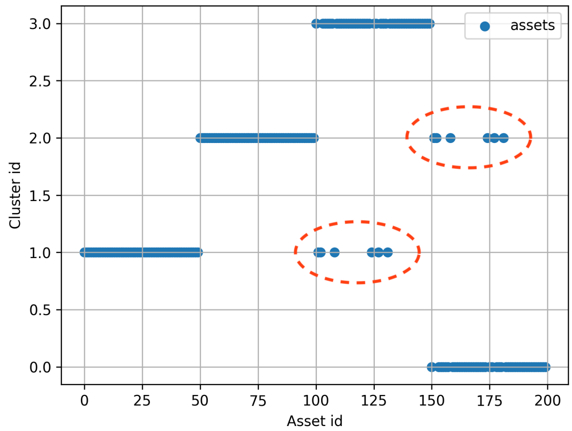
4. **Only the low data assets learn from others** Lastly, a combination of hierarchical and independent modelling was considered in the experiments. This scenario involved clustering and hierarchical modelling similar to the one in case 2. But while all 800 assets here participated in estimating hierarchical model parameters, only those assets belonging to the low data category used the final estimates for classifying the testing dataset. The medium and high data category assets used independent modelling to estimate their Gaussian parameters. Concretely, the final estimates for the assets belonging to the low data category were derived from the hierarchical model, whereas

the final estimates for the assets belonging to the medium and high data category were derived from their independent models.

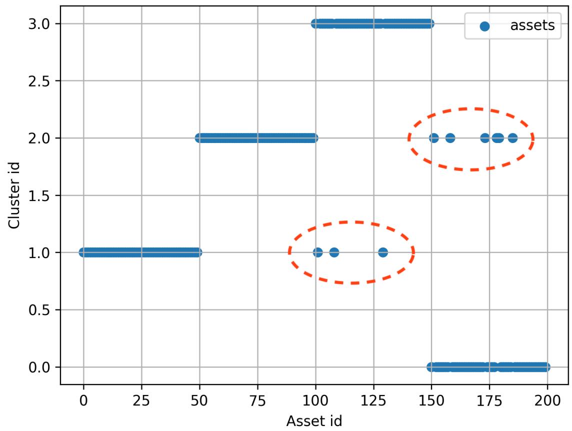
It was observed during the experiments that the accuracy of clustering using EM algorithm relied on the initialisation of parameters, especially the β_k and α_k parameters. These parameters must be initialised such that the algorithm's search space is wide enough and is not trapped in local optima during the early iterations. The approximate initialisations of parameters to ensure a wider search space are mentioned in Section 3.1. However, even with the optimal initialisation, the EM algorithm was unable to cluster the assets due to the wide range of means chosen.

This problem is highlighted in Figure 3.4, where a sample of 50 assets from each of the asset clusters was taken and the total 200 assets thus formed were clustered based on the available 5 and 6 data points only. The figures show both cases- where all assets had the same amount of data, and where the assets are divided into "low", "medium", and "high" data categories explained in Section 3.2.1. In the figures corresponding to the latter case, the assets belonging to the "low", "medium", and "high" data categories are represented in red, orange, and green colours respectively. Also, the number of data points with assets belonging to the low data category were 5 and 6, and were constant for the remaining assets. In all figures, the assets with ids 1 to 50 belonged to the same cluster, 51-100 belonged to the next cluster, and so on. Therefore, these asset ids are expected to be clustered together, which was not the case for only initial 5 or 6 data points. The wrongly clustered assets are marked with the dotted red circle.

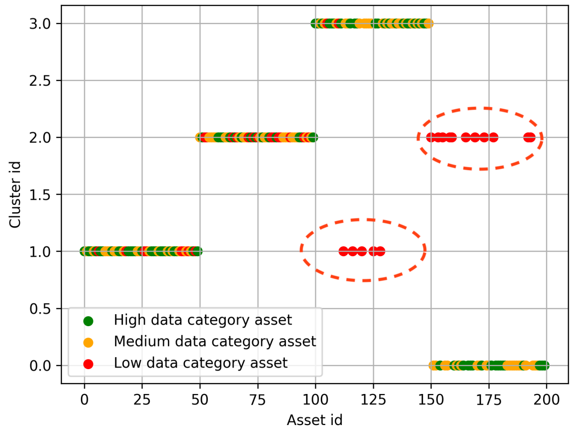
In the real world, this problem can be addressed by including certain categorical data along with the time series data. Categorical data can arise from the operational experience, such as asset's environment, upkeep, operation, etc. However, for the experimental results presented here, it was assured that the assets were correctly clustered in these cases. If it was found that an asset was wrongly clustered, it was manually reassigned to its correct cluster and the results evaluated again. The goal of the experiments is to demonstrate the advantage of hierarchical modelling over the conventional independent modelling on the effectiveness of collaborative learning between assets.



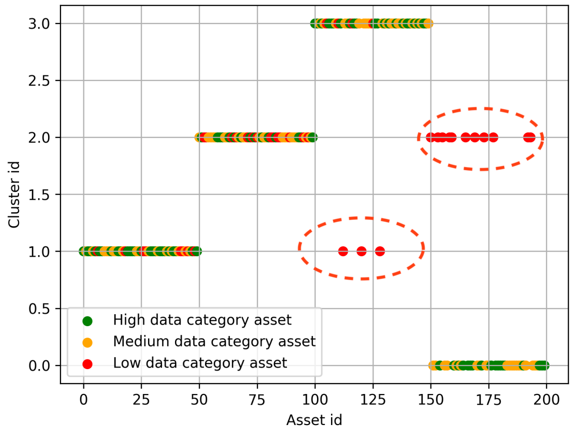
(a) All assets have 5 data points each



(b) All assets have 6 data points each



(c) Low data assets have 5 data points only



(d) Low data assets have 6 data points only

Fig. 3.4 The figures represent the clustering done by the EM algorithm when the assets (low data category assets in (c) and (d)) have 5 and 6 data points only. The incorrectly clustered assets are marked with dotted red circle.

3.2.3 Performance Evaluation

After the estimated model parameters are obtained, the operator must define a region in multidimensional space that encompasses the asset's normal operations data. For the statistical classifiers, this region is often defined based on a critical value from the probability density function (PDF) values, such that any point having the PDF value less than the critical value will lie outside the region and be deemed anomalous. The critical value corresponds to an α significance level, which separates the most likely $100 * \alpha\%$ points from the rest. In other words, the critical value separates $100 * \alpha$ percentile data sampled from the rest.

For the case of multivariate Gaussians, this region is an ellipsoid, and determining its boundary corresponding to the required α level is numerically complex. This is because one cannot simply integrate the tails of the multivariate Gaussian and obtain the boundary corresponding to the required α level. However, for a multivariate Gaussian with dimension d, the squared Mahalanobis distance (D_{md}) of any point with respect to that Gaussian is standard chi-squared with d degrees of freedom[1]. For a standard chi-squared distribution, it is easy to obtain the PDF value separating the most likely $100 * \alpha\%$ points from the rest. This fact can be used to determine if a given data point from the multivariate Gaussian falls within the α level set by the operator or not.

For example, if the α level is set at 0.8, then the corresponding PDF value for a standard chi-squared distribution can be obtained which would in fact be the critical value for the squared D_{md} of the points. Any point having the squared D_{md} greater than the critical value would be deemed anomalous. The $p-$values corresponding to various α levels for a standard 5-dimensional chi-squared distribution are shown in Table 3. These also act as the critical values for the squared D_{md} while generating the ROCs.

Table 3.3 Various α levels used while plotting the ROCs, and the corresponding D_{md} values for the current experiment. These correspond to a standard chi-squared distribution with 5 degrees of freedom

α **level**	$\mathbf{D^2_{md}}$ **value**	α **level**	$\mathbf{D^2_{md}}$ **value**
0.995	0.412	0.5	4.251
0.99	0.554	0.1	9.236
0.975	0.831	0.05	11.071
0.95	1.145	0.025	12.833
0.9	1.61	0.01	15.086
0.75	2.675	0.005	16.75

The squared Mahalanobis distance for any point $\mathbf{X}$ from a given Gaussian distribution with the estimated mean and covariance $\hat{\mu}$ and $\hat{\mathbf{C}}$ is obtained as:

$$D_{md}^2 = (\mathbf{X} - \hat{\mu})^T \hat{\mathbf{C}}^{-1} (\mathbf{X} - \hat{\mu}) \tag{3.27}$$

Areas under the Receiver operator characteristic (ROC) curves were used as the performance metric for comparing hierarchical modelling and with the conventional independent modelling technique. This is a widely used evaluation metric for classification tasks and is

[1] Proof shown in Appendix C

often called the *c-statistic*. It provides an aggregate measure of classification performance across a wide range of α levels.

To plot an ROC, the α levels while classifying the testing dataset were varied across $\{0.995, 0.99, 0.975, 0.95, 0.9, 0.75, 0.5, 0.1, 0.05, 0.025, 0.01, 0.005\}$. An ROC curve was obtained for a single asset and its corresponding testing dataset by plotting the true positive rate (TPR) vs false positive rate (FPR) for each of the alpha levels mentioned above.

Consider a testing dataset with N_P and N_N number of real positive and negative class data points respectively. For the current use case, testing data points sampled from the true underlying distribution were labelled as "negative" class and those sampled from the anomalous distribution were labelled as "positive" class. If a classifier is tested using this dataset and the resulting output comprises of N_{TP} and N_{FP} true positives and false positives respectively, the TPR and FPR are evaluated according to:

$$TPR = \frac{N_{TP}}{N_P} \qquad FPR = \frac{N_{FP}}{N_N} \tag{3.28}$$

The Area Under the ROC Curve (AUC) was used as an indicator of the model's performance for a given asset. From (3.28), it can be observed that a higher AUC is characterised by a high TPR and a low FPR for some α level. A higher AUC means that the classifier is better capable of separating the positive and the negative class in the testing dataset. Therefore, higher the AUC, the better is the classifier. An example ROC for a medium data category asset and its corresponding AUC are shown in Figure 3.5. This ROC was evaluated for the parameters estimated based on hierarchical modelling.

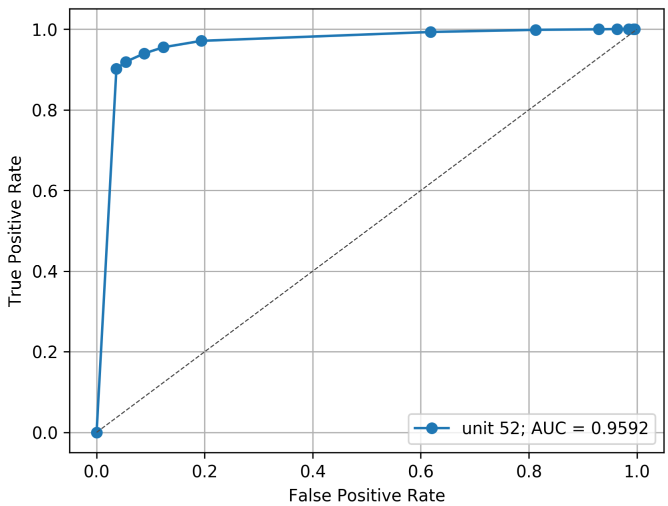


Fig. 3.5 An example ROC for asset id 52 evaluated for testing dataset with l and L equal to 0 and 10 respectively

Such AUCs were evaluated for hierarchical modelling across the fleet and for each testing dataset, and were compared with those obtained using independent modelling.

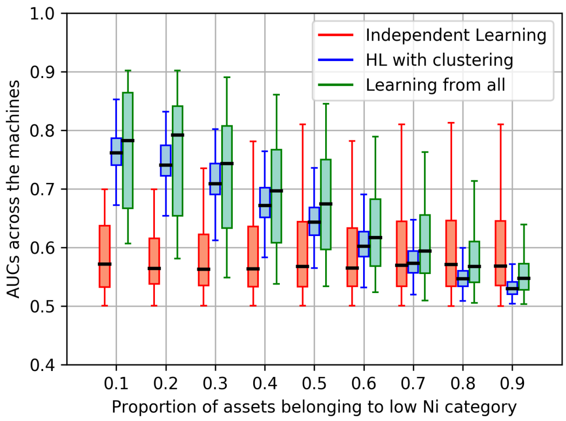
3.2.4 Experimental Results

Using the Areas under the Receiver Operating Characteristic curves as the performance metric

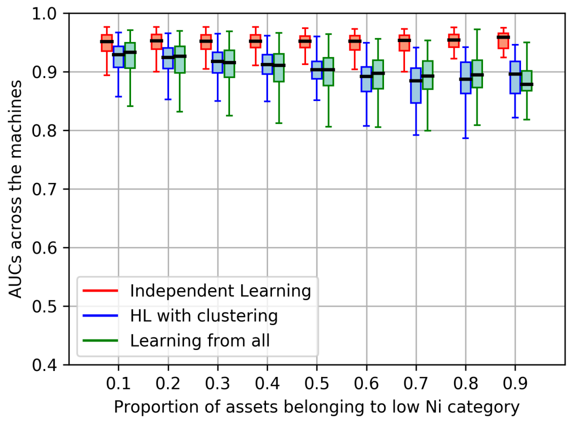
For each of the four scenarios, the AUCs were evaluated for the assets in the fleet as explained in Section 3.2.3. Box plots for each low, medium, and high data category assets for the same testing dataset are shown in Figure 3.6, where “HL” stands for “Hierarchical Learning” where the final estimates are estimated based on the higher level model. Figure 3.6 also includes a combined box plot for all assets in the fleet and for the above described scenarios. These AUCs are presented as box plots. Results corresponding to a subset of test cases are presented here, and the same conclusions hold across all testing datasets. The corresponding testing dataset deviations for all figures are mentioned in their captions.

As an interesting extension to the above described scenarios, the number of data points held by the low data category assets were gradually increased. The number of data points were increased from 5 till 21, so that classifier performances throughout the transition of the assets from low to the medium data category and beyond could be analysed. While doing this, the number of points held by the medium and high data category assets were kept constant at their initial values. Figure 3.7 presents the effect of increasing data at the low data category assets, where 0.2 proportion of assets initially belonged to the low data category. The corresponding testing datasets are mentioned in the sub-captions.

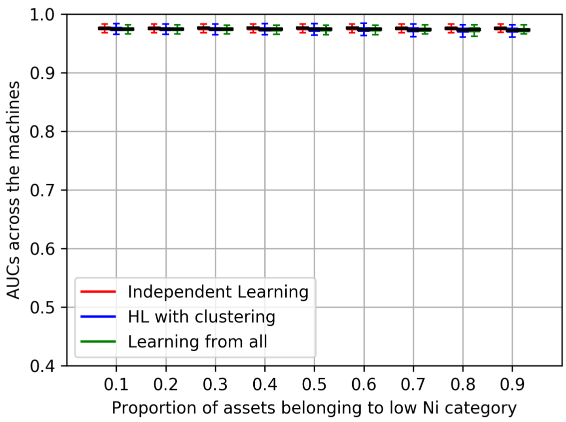
Furthermore, a learning scenario where all 800 assets held the same amount of data was also studied. This was done by simulating the fleet where all assets initially had 5 data points only, which were gradually increased to as high as 500 together across all assets. The classifier performances were studied throughout this transition. Figure 3.8 present the classifier performances when all assets contained the same amount of data. Other results obtained from the experiments described in Section 3.2.2 are presented in Appendix C.1.



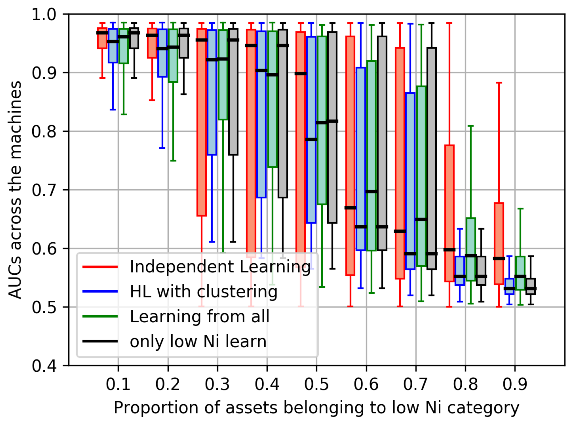
(a) AUCs across low data category assets only.



(b) AUCs across medium data category assets only.



(c) AUCs across high data category assets only.

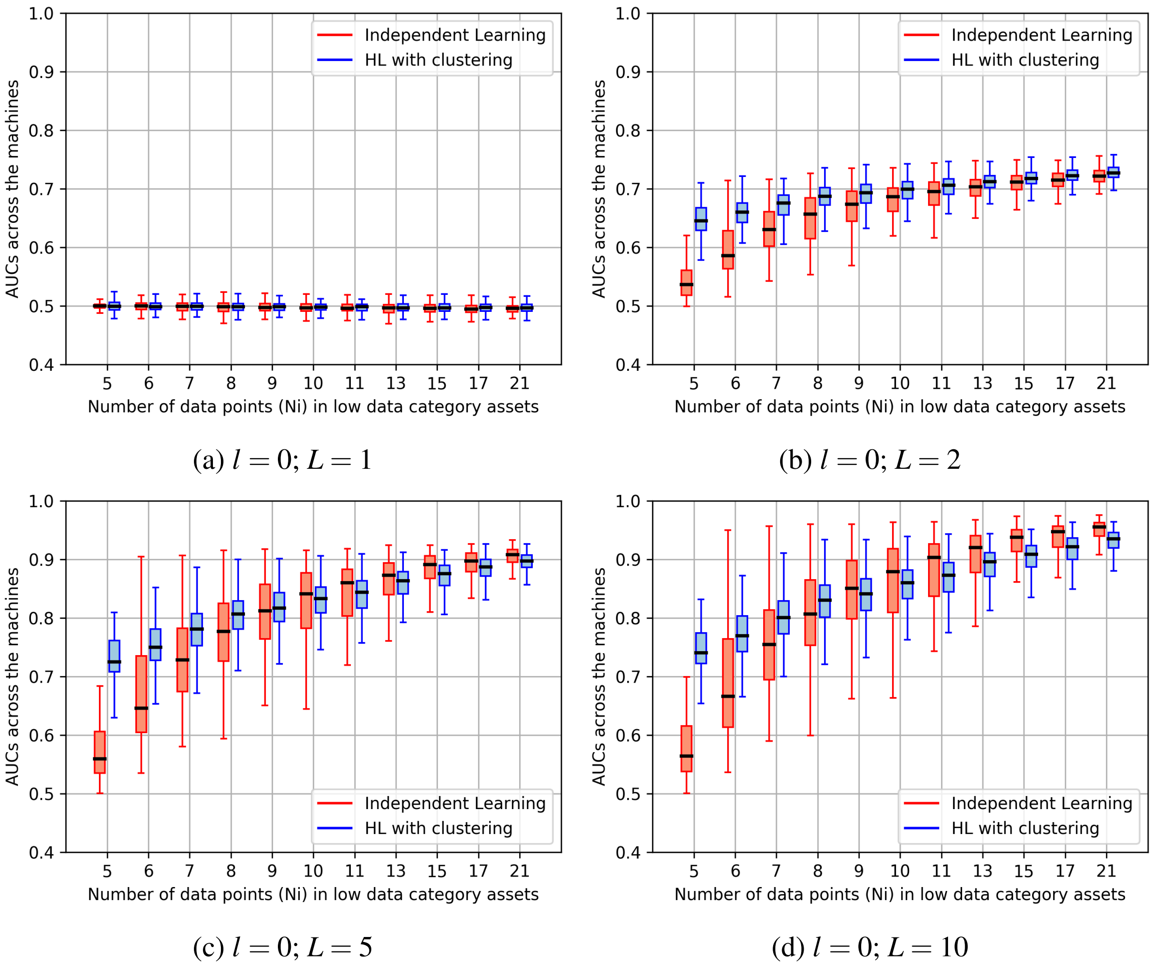


(d) AUCs across all assets, and for all the cases included in the experiments.

Fig. 3.6 Shown here are the AUCs measured for the experiment cases. The subset of assets across which the AUCs are measured are indicated in the corresponding captions. For all the above four plots, the deviation for anomalous data in the testing dataset was set at 1 and 10 for l and L respectively

Figure 3.6 shows that learning from similar assets is more helpful than learning from all assets in the fleet. Learning from all resulted in higher variance in AUCs recorded across all assets, as shown in Figures 3.6a to 3.6c.

The aforementioned points are further highlighted by Figures C.2 and C.3 in Appendix C.1 where the classifier performances for the low data category assets across various testing datasets are presented. In these figures again, the hierarchical model is seen to consistently outperform the independent model, and learning from similar assets shows much lesser variance than learning from all assets in the fleet.

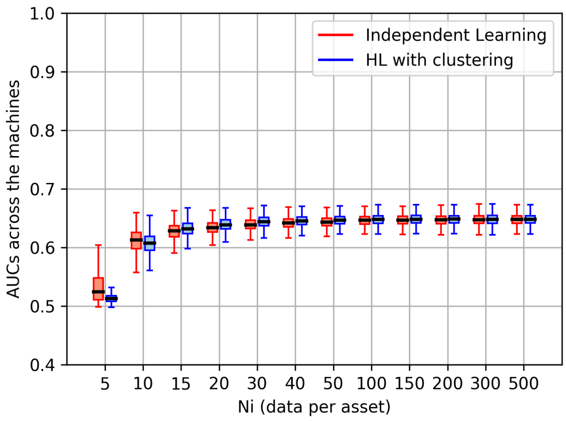


(a) $l = 0; L = 1$ (b) $l = 0; L = 2$

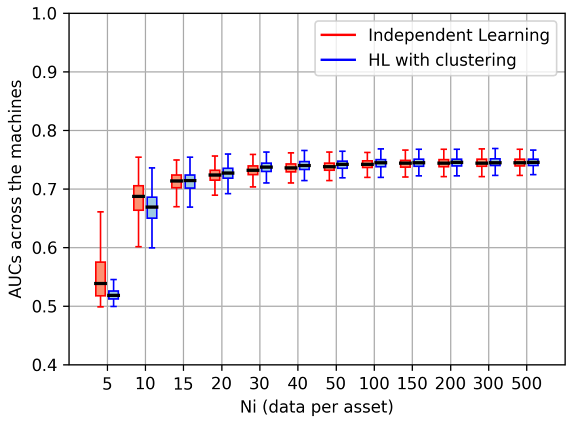
(c) $l = 0; L = 5$ (d) $l = 0; L = 10$

Fig. 3.7 Box plots presenting the effect of gradually increasing data contained by the low data category assets. The captions denote the corresponding deviations in the testing dataset

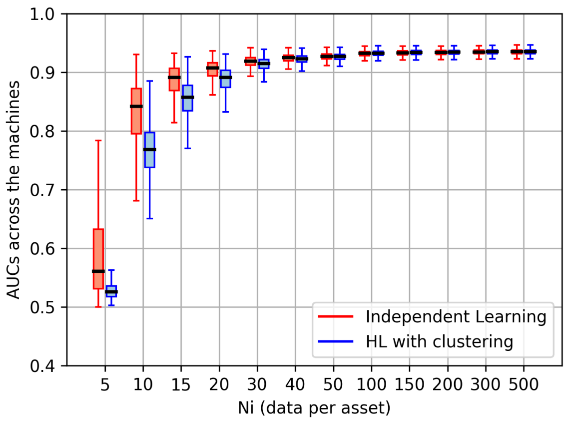
It is observed in Figures 3.6a to 3.6c that hierarchical modelling is beneficial for the assets belonging to the low data category only. For the assets belonging to the low data category, the classifiers obtained using hierarchical modelling show significantly higher AUCs and lower variances than the independent models learning from their own data. This is true especially until the proportion of low data assets in the fleet is less than or equal to 0.6. The same fact is reiterated by Figures 3.7 and 3.9 where until a certain amount of data is accumulated by the asset, it is better for it to rely on hierarchical model estimates. While the threshold corresponds to 13 data points in Figures 3.7 and 3.9, the exact data requirement for the independent model depends on the intra-cluster asset similarities and variance in data, and therefore varies across applications.



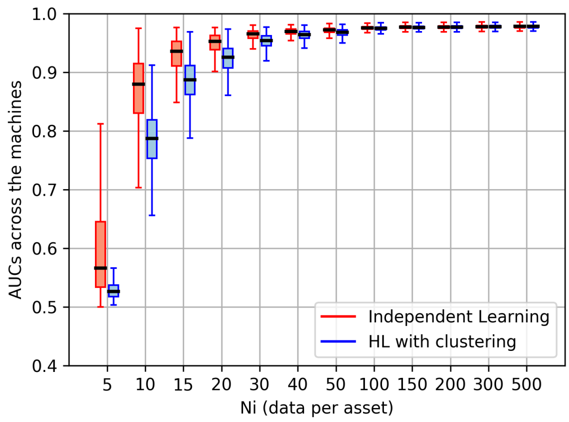
(a) $l = 0; L = 1.5$



(b) $l = 0; L = 2$



(c) $l = 0; L = 5$



(d) $l = 0; L = 10$

Fig. 3.8 Box plots presenting the effect of gradually increasing the data across all assets, when they all had same amount of data. The corresponding testing dataset deviations are denoted in the captions

Figure 3.8 shows that independent modelling is always the better option when all assets in the fleet contain same amount of data. This is true across the entire range from 5 data points until 500 and beyond. But Figure 3.8 also represents that hierarchical modelling eventually converges and becomes similar to independent modelling when the assets keep generating data over time. This confirms our hypothesis that initially the hierarchical model estimates are weighted more towards the general fleet behaviour. The trend seen in Figure 3.8 is an expected outcome because when all assets in the fleet have same amount of data, none of which are clearly indicative of the assets' operating regime. Therefore, the general fleet behaviour, which is a combined behaviour observed across all assets, was not indicative of the correct operating regime as well.

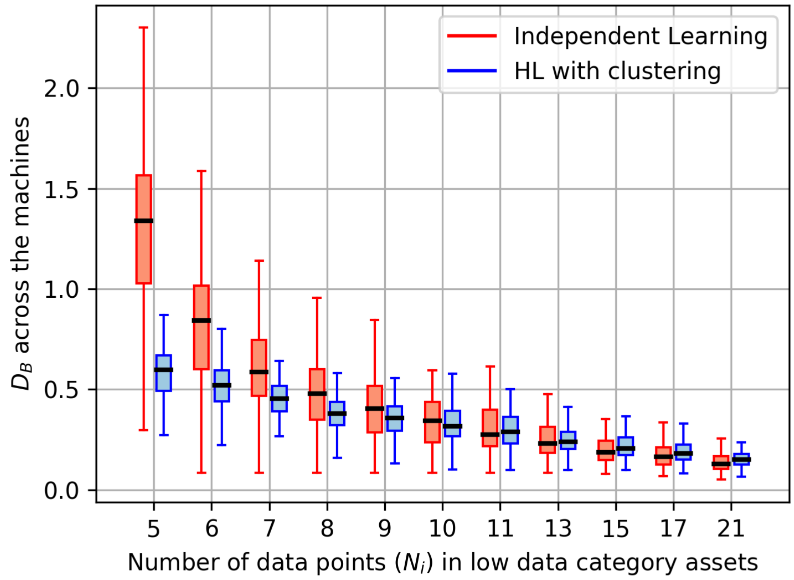
Using the Bhattacharyya Distance as the performance metric

Apart from the performance evaluation metric presented in Section 3.2.3, the Bhattacharyya distance (D_B) was also used to compare the performances of hierarchical and independent asset models.

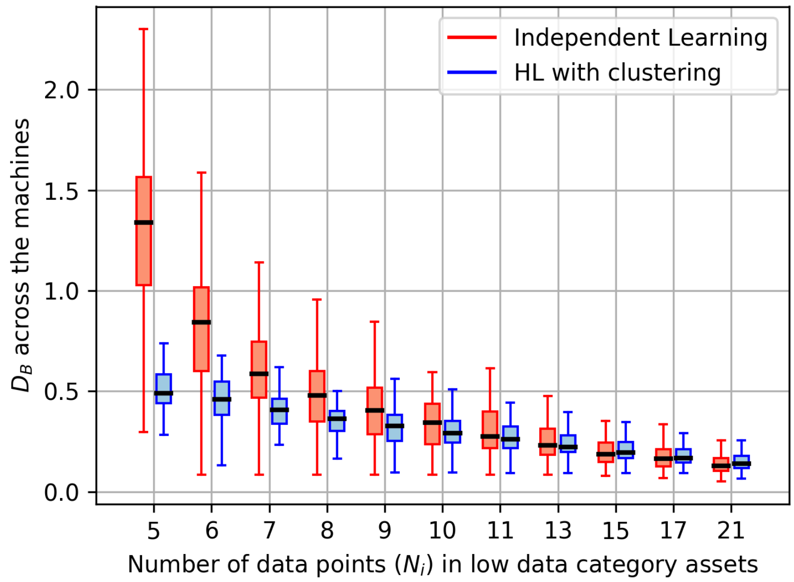
D_B is a distance measure for two multivariate Gaussians, and is calculated according to (3.29) for the Gaussians parameterised by $(\mu_1, \mathbf{C}_1)$ and $(\mu_2, \mathbf{C}_2)$ [10]. A lower value of D_B signifies that the given Gaussians are more similar. For the current application, D_B between the true and estimated Gaussians for all the assets were evaluated.

$$D_B = \frac{1}{8}(\mu_1 - \mu_2)^T \left(\frac{\mathbf{C}_1 + \mathbf{C}_2}{2}\right)^{-1} (\mu_1 - \mu_2) + \frac{1}{2} ln \left(\frac{\mathrm{Det}\left(\frac{\mathbf{C}_1+\mathbf{C}_2}{2}\right)}{\sqrt{\mathrm{Det}(\mathbf{C}_1)\mathrm{Det}(\mathbf{C}_2)}}\right) \tag{3.29}$$

The plots for the evaluated D_B are presented in Figure 3.9. Subfigures 4.10a and 4.10b present D_B evaluated across all assets in the fleet, according to (3.29), as the data points in the low data category assets were sequentially increased. Figure 4.10a corresponds to the case where the range of individual asset means lay within the range $(-25, 25)$ and $(275, 325)$ for the two model types. Figure 4.10b corresponds to the narrower range of means $(-5, 5)$ and $(295, 305)$ for the two model types. Covariances used to represent the asset operating conditions were the same for both figures and mentioned in (3.25 and 3.26). The results presented in Figure 3.9 correspond to the same experimental setup as for Figure 3.7.



(a) Asset means $\in (-25, 25)$ and $(275, 325)$



(b) Asset means $\in (-5, 5)$ and $(295, 305)$

Fig. 3.9 Box plots presenting the D_B recorded across the assets belonging to the low data category. A lower value of D_B signifies that the given Gaussians are more similar

It was observed that the performance of hierarchical model was affected by the choice of range of means mentioned in 3.2.1. A shorter range of means would signify that the assets were more similar to one another, resulting in an improved performance of the hierarchical model. This fact can be observed from the results from the same experiment with shorter ranges of means, $(-5,5)$ and $(295,305)$, presented in Appendix C.2 and Figure 3.9 for both performance metrics.

3.3 Conclusions

The framework of hierarchical model is proposed here as a systematic technique for the similar assets within a fleet to collaboratively learn from one another, and improve the performances of their statistical classifiers for anomaly detection. The asset condition monitoring data are modelled using multivariate Gaussians. But the hierarchical model, unlike conventional maximum likelihood estimation, involves higher level distributions from which the asset level Gaussian parameters are sampled. The higher level distributions are shared by the clusters of similar assets, where similarities arise by the virtue of the assets operating in similar conditions or being of the same model type. The higher level distributions for the covariances and the means of the asset level Gaussians are modelled using their conjugates, i.e. Inverse-Wishart and Gaussian respectively.

A better classifier for a given asset and a testing dataset is characterised by a higher AUC, and a lower D_B. However, while analysing the performance of that classifier across the entire fleet, its consistency also plays a key role. An operator would prefer having a classifier showing consistent but slightly worse performance rather than an unreliable classifier which shows high AUC for some assets in the fleet but low for others. Comparing the Bhattacharyya distance for the two techniques, it can be concluded that hierarchical modelling significantly improves the performances of conventional classifiers in the early periods of asset operations. This is the period when sufficient training data are not available to estimate the Gaussian parameters using maximum likelihood methods. The higher level distributions are also representative of the general behaviour of the asset fleet, that is of interest to the operators who want an overall understanding the fleet performance.

Moreover, an important conclusion in this chapter is that when all the assets in a fleet have sparse data, it is better to rely on the independent models rather than the hierarchical model. This observation can be noted in Figure 3.8.

Chapter 4

Collaborative Prognosis using a Statistical Hierarchical Model

This chapter contributes towards addressing research questions 1 and 2 and achieving research objectives 4 and 5 outlined in Chapter 1, and is divided into two parts.

The first part presents a statistical hierarchical model for modelling the times-to-failures observed in a fleet comprising of sub-fleets, or clusters, of similarly deteriorating assets. Similar to the hierarchical model for anomaly detection presented in Chapter 3, the hierarchical model for modelling the times-to-failures mitigates the problem of high variance encountered while independently modelling the times-to-failures observed in the asset clusters with sparse data. Given its popularity in Reliability Engineering, a hierarchical model of Weibull density functions is shown in this chapter as an example. It should be noted that distributions other than Weibull shall also be used to model the times-to-failures.

The second part describes the procedure for real-time collaborative prognosis using the proposed hierarchical Weibull model, and demonstrates it for a fleet of simulated turbofans. This represents a scenario where the observed failures and the corresponding failure trajectories are used as the *training dataset* for estimating the Remaining Useful Lives (RULs) of the operating assets part way through their failure trajectories, which constitute the *testing dataset*. Given the observed failures, the experiments presented herewith analyse the prediction accuracies along the life of an operating asset and also the effect of clustering on the prediction accuracy.

The following chapter is structured as follows: the simulated turbofans dataset used for the experiments is described in Section 4.1. Section 4.2 describes the simulation of real-time asset operations using failure trajectories using the simulated dataset. Sections 4.3 and 4.4 describe the steps to preprocess and cluster the turbofan data respectively, which are used for analyses in the later sections. The problem of estimating the Weibull density function using the fleet-wide and independent cluster-specific data is discussed in Section 4.5. The proposed hierarchical model is described and its implementation to model the times-to-failures is presented in Section 4.6. Section 4.7 presents the results from the experiments that were conducted to analyse the efficacy of the hierarchical model. Section 4.8 describes the procedure to estimate the RUL of an operating asset, which is then implemented for prognosis of the simulated asset operations in Section 4.9. Section 4.9 subsequently presents an experiment conducted to analyse the effect of clustering on collaborative prognosis using the hierarchical Weibull model. The conclusions are summarised in Section 4.10.

4.1 Example Dataset Description

This section introduces the publicly available Turbofan Engine Degradation Simulation dataset used for the experiments discussed herewith. This dataset was generated using a Matlab based simulation software called Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) [159]. It is subsequently referred as the C-MAPSS dataset.

C-MAPSS software is capable of simulating turbofan engines operating in various user-defined conditions. The conditions include the altitude at which the engine is operating, its Mach number, and the temperature at sea-level. Thermodynamic equations are used to calculate fluid flow parameters, and the health conditions of the engines are reflected in their sensor measurements from various internal locations. The simulated turbofans are each monitored using 21 sensors [160].

The turbofans comprise of independent sub-systems including regulators, limiters and control systems. The limiters resemble the warning-trip mechanisms typically present in the industrial turbo-machinery that prevent the machines from exceeding their pre-set tolerances. In C-MAPSS, there are limiters for the core speed, the engine-pressure ratio, for the high pressure turbine exit temperature, and for the static temperature at the high-pressure compressor. An engine is deemed inoperable/ failed when any of the limiters are exceeded [160].

The C-MAPSS dataset represents a fleet of turbofans with continuously degrading health since the start of their operations, until they all eventually fail. A turbofan's degradation is manifested in the simulations as the percentage reduction in the constituting component's efficiency($e(t)$) and flow($f(t)$) values at timestep (t) compared to those at its healthy state (at timestep $t = 0$). The overall health index of a machine at time t is a combined function of flow and efficiency of the overall engine: $H(t) = g(f(t), e(t))$.

The $e(t)$ and $f(t)$ values of a given component are simulated to degrade with time according to an inverse exponentially decreasing function. However, no two simulated

Table 4.1 A Sample of FD_001 Dataset

AssetID	Cycles	OC1	OC2	OC3	s1	...	s21
1	1	-0.0007	-0.0004	100	519	...	23.419
1	2	0.0019	-0.0003	100	519	...	23.424
...	...	...	...	...	...	...	...
2	1	-0.0018	0.0006	100	519	...	23.458
...	...	...	...	...	...	...	...
2	287	-0.0005	0.0006	100	519	...	23.084
...	...	...	...	...	...	...	...
100	200	-0.0032	-0.0005	100	519	...	23.052

turbofans are identical because the parameters governing inverse exponential function are randomly chosen from their corresponding permissible ranges of values. The turbofans also commence operation with a slight but random initial deterioration to replicate the real world manufacturing inefficiencies, and noise was added to the sensor measurements to replicate the real world errors. Comprehensive information about the simulator can be found in [160].

As a result of a turbofan's health degradation, the fluid flow parameters recorded by sensors across various components deviate from their healthy operation regimes. During the simulations, the time series of sensor measurements ranging from a given turbofan's healthy state until its failure are recorded with their corresponding cycles of operations and operating conditions. The failure trajectories generated by the C-MAPSS software are analogous to the real-world trajectories.

The C-MAPSS dataset is divided into four files, each comprising of failure trajectories for turbofans operating in various conditions and incipient failure modes. Files FD_001 and FD_003 are used for the experiments discussed herewith and comprise of total 200 turbofans operating at sea-level conditions. All turbofans represented in FD_001 were simulated to fail because of their high-pressure compressor degradation, and turbofans in FD_003 could fail either due to high-pressure compressor degradation or fan degradation. The data shown in Table 4.1 as an example is sampled from the FD_001 file where the columns correspond, from left to right, to the turbofan ID, cycle of operation, three parameters describing the operating conditions, followed by the 21 sensor values recorded at every cycle of operation. The following Sections 4.3 and 4.4 describe the steps for preprocessing and identifying the clusters of similarly deteriorating turbofans in the C-MAPSS dataset respectively.

4.2 Simulating Real-time Asset Operations

Train_FD001 and train_FD003 files from the C-MAPSS dataset jointly contain failure trajectories of 200 simulated turbofans operating in sea-level conditions and that fail over a period of time in either of the two failure modes: high pressure compressor failure, or fan degradation. The failure trajectories are in the form of time-series of condition data recorded from the turbofans' healthy state until failures, represented here as $\mathbf{x}_n^{0:t_n^f}$ for a single turbofan n where t_n^f represents the time at which the turbofan n failed. The condition data recorded at each time-step comprises of a vector $\mathbf{x}_n$ of sensor values measuring the internal/ external operating parameters of the turbofan.

170 simulated turbofans from the failure trajectories obtained after preprocessing were used as the training dataset while the remaining 30 were used as the testing dataset. Complete time-series of the condition data were available from the 170 turbofans comprising the training datasets, whereas the remaining 30 were simulated to be operating in real-time such that only part of their time-series were available for RUL predictions. Concretely, for the n^{th} turbofan in the testing dataset operating until t_n time-steps, only the $\mathbf{x}_n^{0:t_n}$ datapoints from its failure trajectory were used for predictions.

4.3 Preprocessing the Dataset

This section describes the steps to preprocess the C-MAPSS dataset. For the experiments discussed herewith, preprocessing the C-MAPSS dataset involved the following steps:

1. **Removing the noise:** Since a mixture of Gaussian white noise is added to the sensor measurements in the simulator, a rolling mean for a window of 20 cycles was evaluated for each sensor and turbofan across the fleet to clean the data.

2. **Normalising the data:** Normalising the data is particularly important for the assets such as turbofans because the sensors measure diverse thermodynamic parameters. These parameters are associated with units and sensor measurements that vary over a wide range of values. In the case of C-MAPSS dataset, the thermodynamic quantities corresponding to the sensor indicators, obtained from the introductory paper [160], are represented in Table 4.2. The diversity of thermodynamic parameters and their quantitative ranges is clear form Tables 4.2 and 4.1. The measurements for every sensor were therefore normalised across the fleet.

3. **Removing the redundant sensors:** This was done to minimise the dimension of the input. The redundant sensors were identified as the ones that did not show any deviation

Table 4.2 Sensor IDs and the corresponding thermodynamic quantities measured in the C-MAPSS fleet. This table is obtained from [160], and the sensors considered for analysis after preprocessing the dataset are indicated in bold text.

Sensor ID	*Description of the measured quantity*	*Unit of measurement*
s1	Total temperature at fan inlet	°R
s2	**Total temperature at LPC outlet**	**°R**
s3	**Total temperature at HPC outlet**	**°R**
s4	**Total temperature at LPT outlet**	**°R**
s5	Pressure at fan inlet	psia
s6	Total pressure in bypass-duct	psia
s7	**Total pressure at HPC outlet**	**psia**
s8	Physical fan speed	rpm
s9	Physical core speed	rpm
s10	Engine pressure ratio (P50/P2)	–
s11	**Static pressure at HPC outlet**	**psia**
s12	**Ratio of fuel flow to Ps30**	**pps/psi**
s13	Corrected fan speed	rpm
s14	Corrected core speed	rpm
s15	**Bypass Ratio**	**–**
s16	Burner fuel-air ratio	–
s17	**Bleed Enthalpy**	**–**
s18	Demanded fan speed	rpm
s19	Demanded corrected fan speed	rpm
s20	**HPT coolant bleed**	**lbm/s**
s21	**LPT coolant bleed**	**lbm/s**

in the measurements throughout the failure trajectories. Moreover, the sensors showing no predictive behaviour were also manually identified and removed from the analysis. Figure 4.1 shows the normalised and clean sensor values for a randomly selected failure trajectory/ asset from the fleet. The sensors showing no deviations are marked in red. The set of redundant sensors were different for the two failure modes present in the fleet, therefore the sensors comprising an intersection of the two sets of the redundant sensors were eliminated. The sensors that were selected for analyses are marked in Table 4.2 in bold text.

4.4 Clustering for the C-MAPSS dataset

This section describes the process of clustering the turbofans comprising the C-MAPSS dataset. Assets comprising a cluster are expected to be homogeneous in the sense they commence operations in a similar health state, and undergo similar rate of deterioration. Unlike jointly clustering and estimating the hierarchical model parameters for anomaly detection, the clustering and the model parameters estimations are treated herewith independently and sequential steps. This is because for collaborative prognosis the clusters of similarly deteriorating assets need to be identified by comparing the failures trajectories comprising of the time-series of asset condition data. Often there exist categorical indicators such as make type, operators, applications, etc. that enable clustering at a higher level. In the case of C-MAPSS dataset, since there are no such indicators to identify the asset clusters, only the time-series asset condition data is used herewith to identify the clusters of similarly deteriorating turbofans.

The turbofans comprising the C-MAPSS dataset commence their operations with a random level of degradation in healthy range (not equivalent to failure), and continuously deteriorate at different rates. As such, the lengths of failure trajectories span over a wide range as shown in Figure 4.2. The diversity in initial health state of the turbofans and the deterioration rates are also reflected in the sensor values throughout the failure trajectories. As an example, the values of sensor *s15* for a random sample of 15 turbofans are shown in Figure 4.3a.

The diversity in degradation behaviours observed in the C-MAPSS data is furthermore apparent in their hazard and log-hazard plots, shown in Figure 4.4. Hazard ($\lambda(t)$) of an asset at time t is defined as the instantaneous rate of failure. It is the probability of an asset failing at time t, given that it has survived until time t. Hazard of a given asset at time ($\lambda(t)$) can be mathematically expressed as:

$$\lambda(t) = \frac{P(t \leq T < t + dt | T \geq t)}{dt} \tag{4.1}$$

For the plot shown in Figure 4.4, $\lambda(t)$ it is calculated as the fraction of turbofans failing in a given time interval to the number of turbofans that survived until that interval. The interval in the case of C-MAPSS dataset was 1 cycle, and hazard was calculated for time until all turbofan failures were observed. The intervals for which there were no failures were associated with hazard 0, and were ignored while transforming the plot into log-hazard. Figure 4.4 indicate the existence of asset clusters following diverse increasing hazard trends (marked in the log-Hazard plot, where it is clearer).

Figure 4.4 amplifies the diversity in failure behaviours across the C-MAPSS dataset. Figure 4.4b especially shows distinct, nearly linear, functions that give a direct indication of presence of at least 5 clusters. The effect of the presence of sub-fleets of similarly deteriorating assets on the empirical log-hazard curve is described with an experiment in Section 4.4.1.

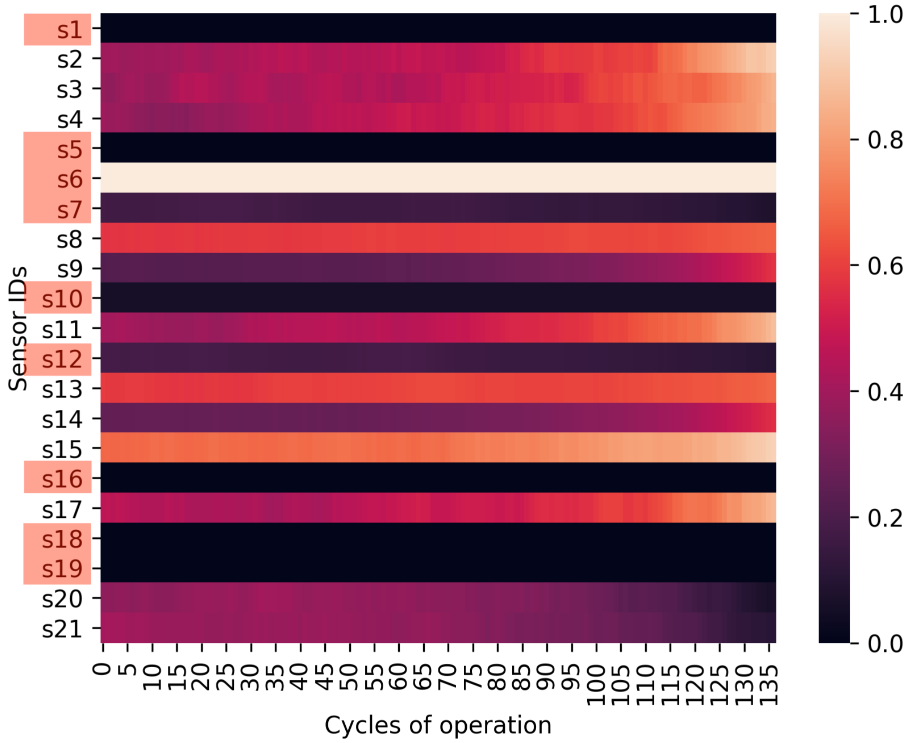


Fig. 4.1 Clean and normalised sensor values for a randomly selected failure trajectory/ asset (asset ID 27) from the fleet. The sensors showing no deviation are highlighted in red.

The given number of sensors, lengths of failure trajectories, and number of turbofans renders the comparison of multivariate time-series a non-trivial problem. To enable the comparison, quadratic polynomials were fit to the time-series of sensor values ranging from the initial till the failed state of the turbofans, for every sensor and turbofan.

The coefficients corresponding to the polynomial fits effectively quantify the rate of deterioration and the initial health state of the turbofans. Moreover, the dimensions are consistent and irrespective of the lengths of the trajectories. This greatly reduces the dimensionality of the data while retaining the necessary information. The polynomials were fit to the normalised sensor values, and normalised trajectory lengths for the turbofans comprising the fleet. Figure 4.3b shows an example of a quadratic polynomial fit for the sensor s15 values corresponding to the same set of turbofans shown in Figure 4.3a.

Figure 4.3b shows that fitting polynomials to the constituent sensor data of a failure trajectory is an efficient technique to represent the underlying information about the asset's degradation behaviour. This is especially useful while using independent algorithms for clustering and modelling the observed failures.

In the transformed dataset thus obtained, an array of sensor values was reduced to three coefficients of the corresponding fitted polynomial. The transformed failure trajectory of each turbofan was represented by total 10×3 features. Principal component analysis (PCA) was further implemented, and the first 10 components were chosen for clustering the turbofans. The first 10 components explain upto 99.72% of the variance in original data.

It should be noted that PCA forms the part of the data preprocessing step in the analytics pipeline, whereas the explainability refers to the proposed collaborative learning technique of the statistical hierarchical modelling. The claim that statistical hierarchical modelling is advantageous in terms of explainability refers to the explainability of the model parameters. The higher level parameters represent the overall fleet behaviour, whereas the lower level parameters represent the individual asset behaviours. Explainability should not be confused here with causality, where the root cause of the failure can be traced back as feedback to the operator.

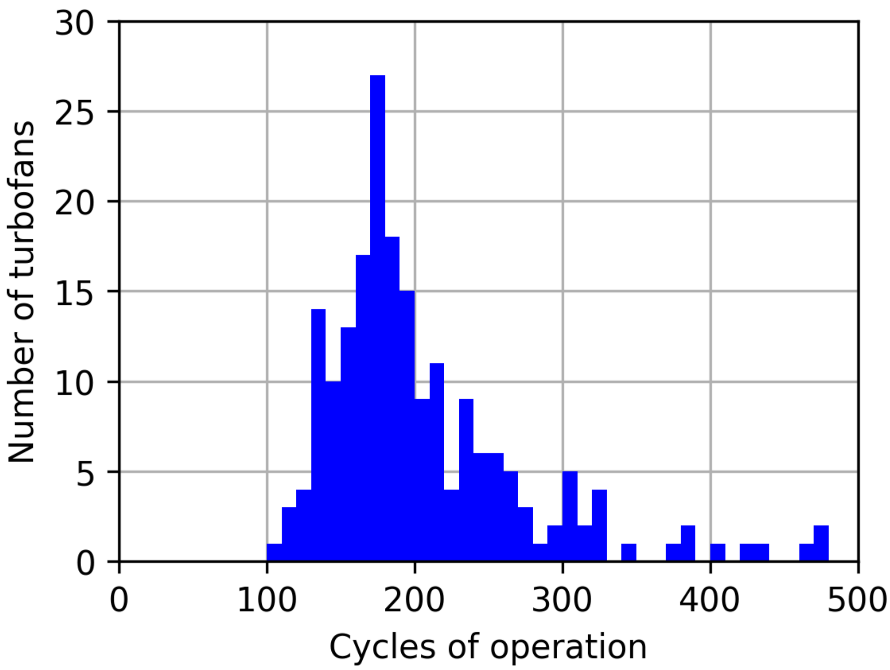
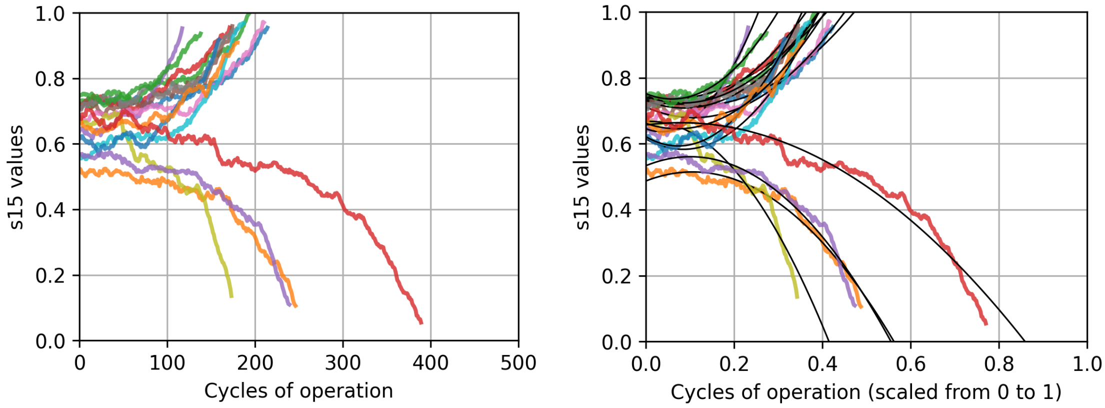


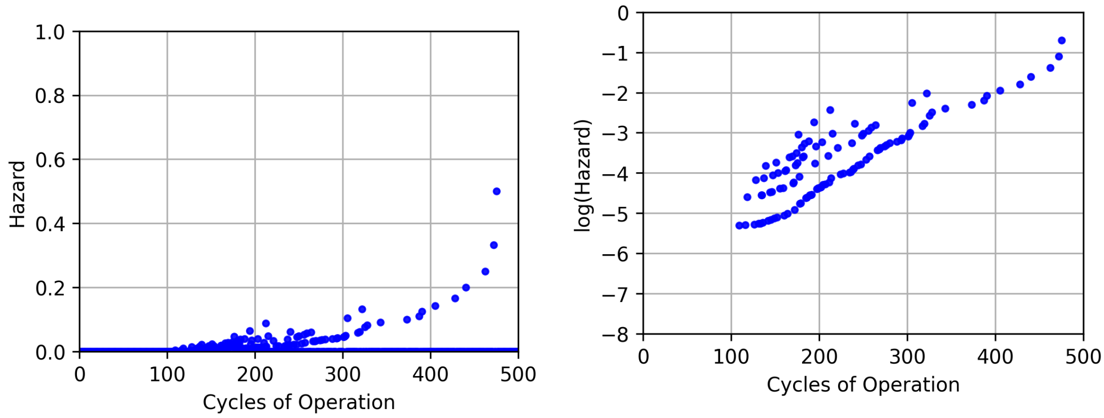
Fig. 4.2 Histogram showing the range of failure trajectory lengths observed in the C-MAPSS dataset.



(a) The values of sensor 's15' for a random sample of 15 turbofans, each represented by a different colour.

(b) Quadratic polynomials fitted to the same subset of turbofans. The fitted quadratic polynomials are represented in solid black lines.

Fig. 4.3 The trends in 's15' values across a random set of turbofans, and the quadratic polynomials fitted to the same set of values.



(a) A plot showing the turbofan hazards across the entire timeline until all turbofans have failed.

(b) A plot showing the logarithm of turbofan hazards across the entire timeline until all turbofans have failed.

Fig. 4.4 Hazard and log-Hazard evaluated at every time-step for the C-MAPSS fleet.

Finally, the k-means clustering algorithm was implemented for clustering the transformed coefficients corresponding to the turbofans. To identify optimal number of clusters, number of clusters for k-means were iteratively increased until a new cluster caused a reduction in the total inertia (sum of squared distances from the cluster centroids) by less than 10%. Total 11 clusters were obtained for the C-MAPSS fleet, each comprising of 37, 35, 30, 21, 17, 16, 11, 11, 9, 8, and 5 turbofans.

The flowchart in Figure 4.6 describes the steps involved in preprocessing and clustering the C-MAPSS dataset. The steps preprocessing and clustering the C-MAPSS dataset are also summarised in the pseudocode presented in Algorithms 2 and 3 respectively.

Algorithm 2: To preprocess the C-MAPSS dataset

Result: Processed failure trajectories.

for *each turbofan i* **do**
  $t = 0$
  while $t < (t_i^f - 20)$ **do**
    $\mathbf{x}_t^i = \frac{\sum_t^{t+20} \mathbf{x}_t^i}{20}$
    $t = t + 1$
  end
end
for *each parameter* $\mathbf{x}$ **do**
  $\mathbf{x} = \frac{\mathbf{x} - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})}$
  if *var*($\mathbf{x}$) *= 0* ***or*** $\mathbf{x} \in$ *redundant sensor* **then**
    drop($\mathbf{x}$)
  end
end
return: *Processed failure trajectories.*

Algorithm 3: To cluster the similarly deteriorating turbofans comprising the C-MAPSS dataset

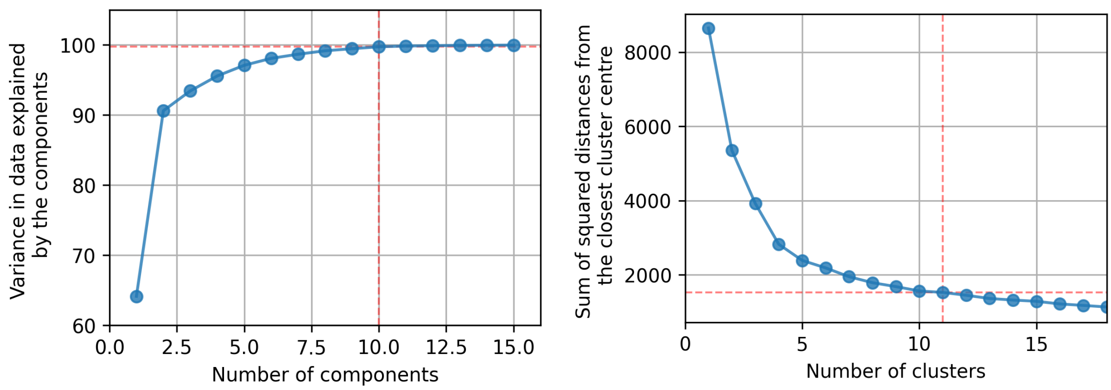
Result: The cluster ID for each simulated turbofan.

for *each asset i* **do**
  for *each parameter* $\mathbf{x}^i$ **do**
    $\mathbf{x}_{cff}^i = fitQuad(\mathbf{x}_{0:t_i^f}^i)$
  end
  $\mathbf{X}_{pca}^i = PCA(\mathbf{X}_{cff}^i, 10)$
end
$k = 2$
while *True* **do**
  inertiaRed = $\frac{inertia(k+1)}{inertia(k)}$
  if *inertiaRed < 0.1* **then**
    cluster ID $= kMeans(\mathbf{X}_{pca}, k)$
    break
  end
  $k = k + 1$
end
return: The cluster ID for each simulated turbofan.

Referring to the Algorithms 2 and 3, the fleet comprises of total I turbofans with each indexed as i. The failure time of turbofan i is represented as t_i^f. The set of condition parameters recorded for the turbofan i at time t is represented as $\mathbf{x}_t^i$ and the same across the entire fleet is represented as $\mathbf{x}$. Functions $max()$,$min()$ and $var()$ are used to calculate the maximum, minimum, and the variance of the corresponding condition parameter. The functions $fitQuad()$, $PCA()$, and $kMeans()$ are used for fitting the quadratic polynomial to return the coefficients, implementing the PCA to return the components, and evaluating the clusters using the k-means algorithm to return the cluster IDs respectively, with the corresponding values of components chosen/ the number of components indicated as the inputs. The failure trajectory of a turbofan i is represented as $\mathbf{X}^i$ and across all the turbofans is represented as $\mathbf{X}$.

Figure 4.7 shows the log-hazard plots across the clusters, which is plotted using the same method as for the plot in Figure 4.4. The cluster IDs are also indicated in Figure 4.7 for reference in the following text in this chapter. It can be observed in Figure 4.4 that the log-hazard plots span across a narrower range and comprise of singular functions, unlike the fleet-wide log-hazard plot in Figure 4.4.

Figure 4.8 shows the values of sensor *s15* in failure trajectories across the clusters. The diversity in failure rates and starting condition across the clusters is apparent in the figure. Moreover, the two different failure modes in the fleet are indicated by the *upward* vs *downward* trends in the sensors values. The clustering algorithm clearly separates these two, and further identifies the sub-clusters beyond the basis of just the failure modes.



(a) Percentage of variance in data explained vs the number of PCA components chosen.

(b) Reduction in inertia for every added cluster in the k-means algorithm.

Fig. 4.5 Reason for choosing the first 7 principle components, and setting the number of clusters to 7 in the analyses.

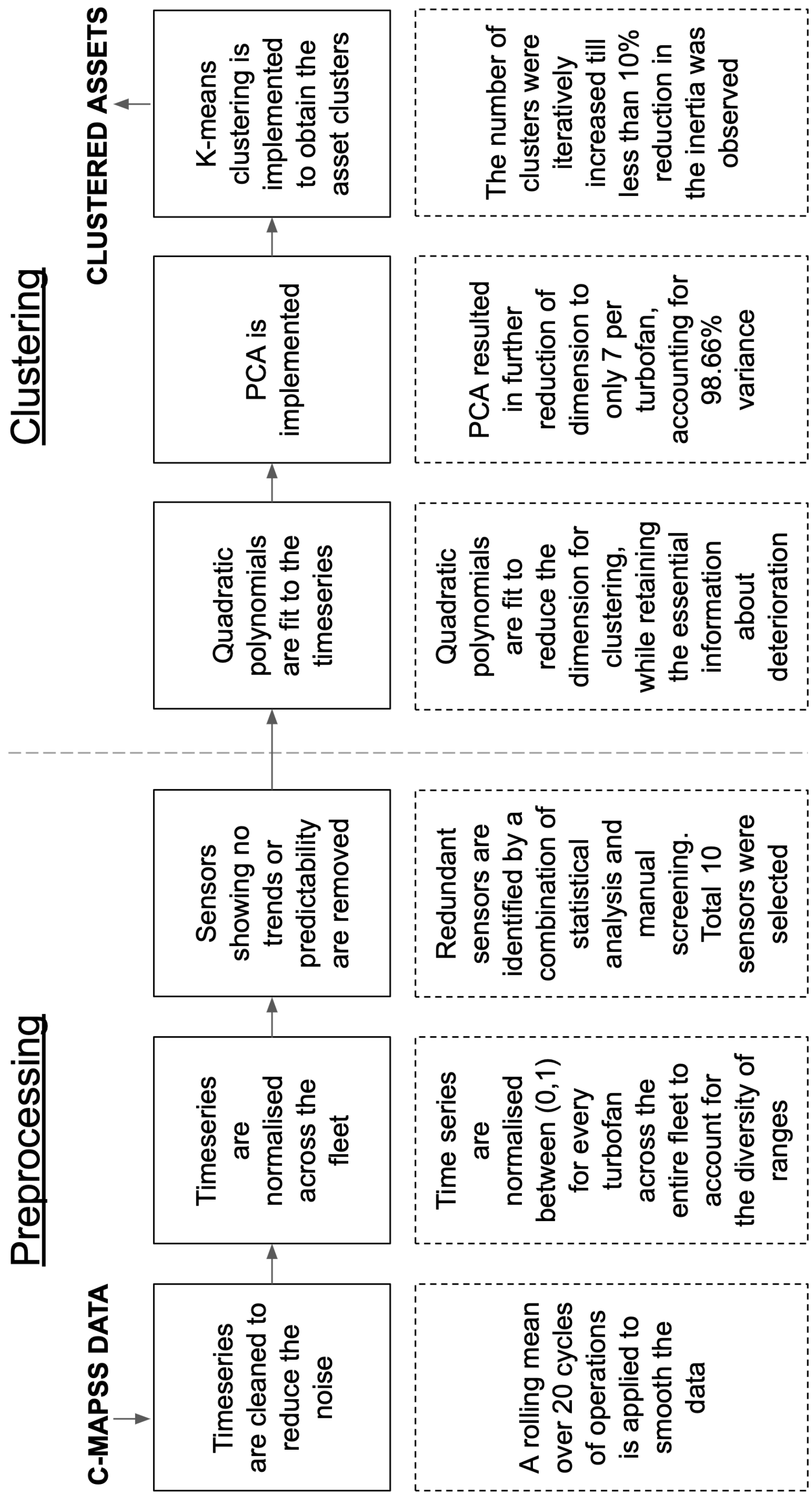


Fig. 4.6 Flowchart describing the steps to preprocess and cluster the C-MAPSS data.

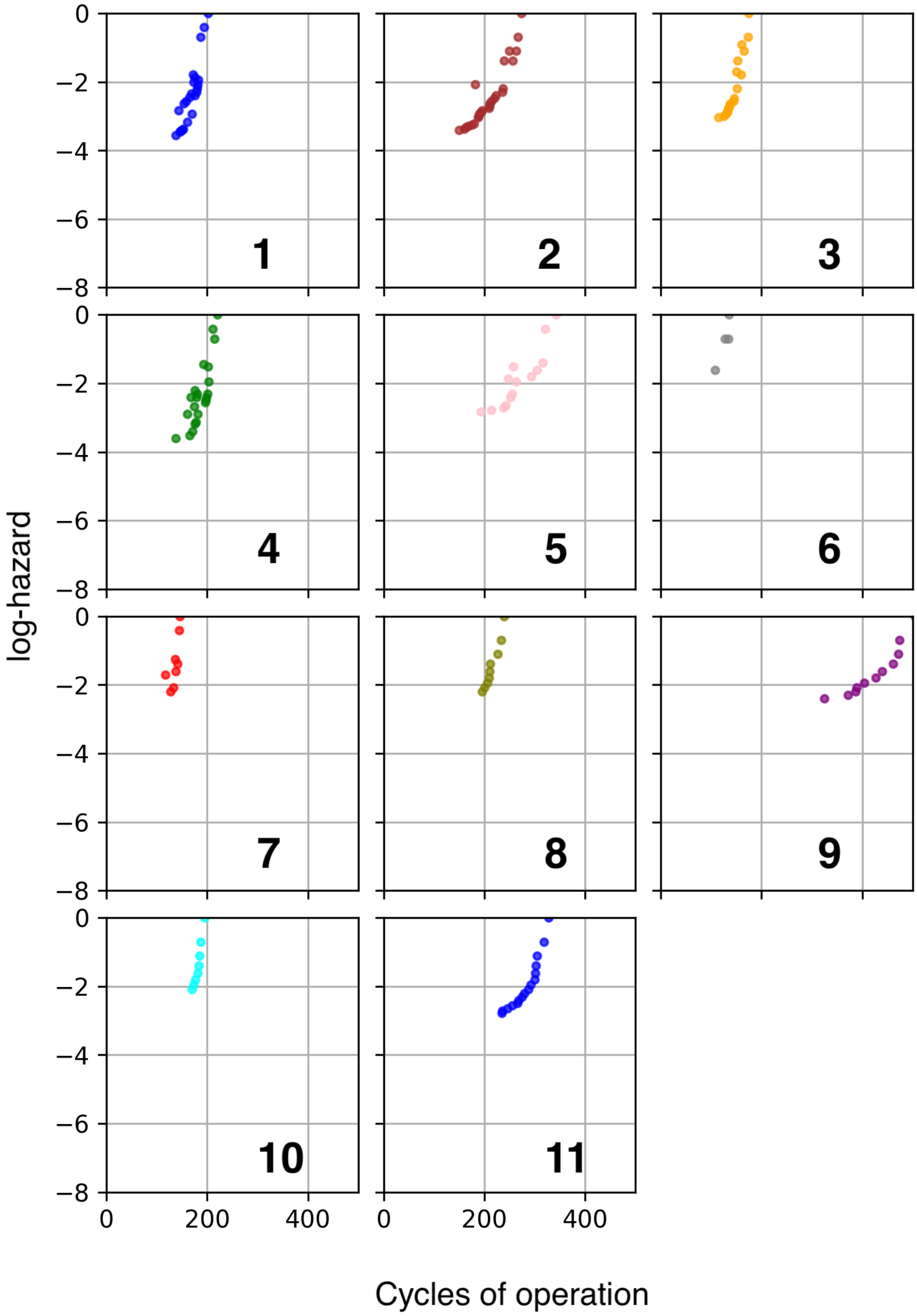


Fig. 4.7 Values of sensor *s15* in failure trajectories across the clusters, each represented in a separate colour. The same colour code to represent the clusters is followed in this chapter.

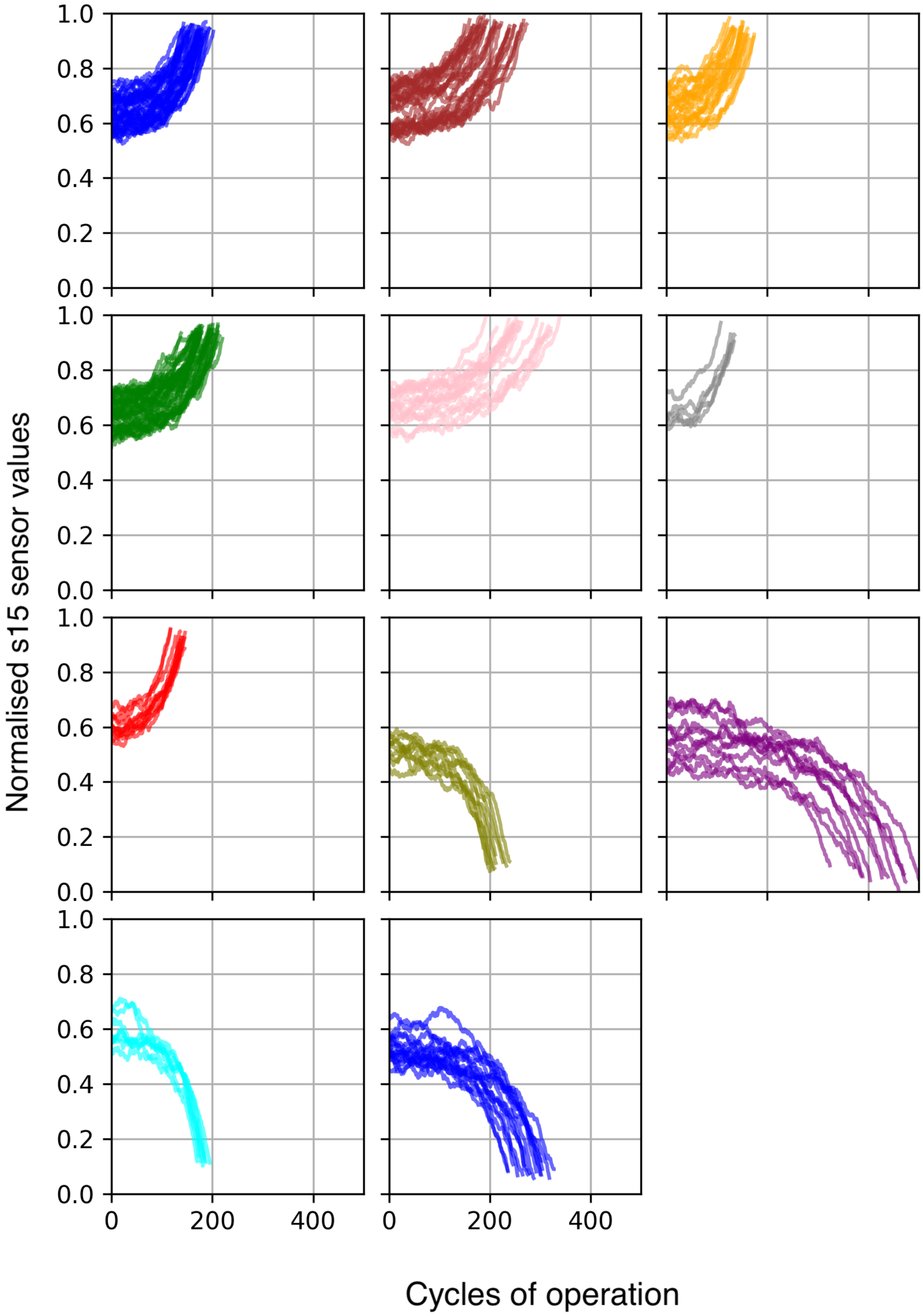


Fig. 4.8 Values of sensor *s15* in failure trajectories across the clusters.

4.4.1 On the Effect of Sub-fleets on the Empirical Log-Hazard Plot

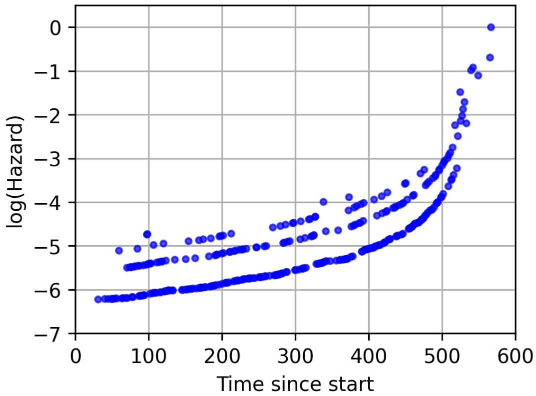
The effect of presence of sub-fleets on the empirical log-hazard curve is described herewith using a simple experiment. Times-to-failures in four fleets of 500 assets each with varying diversity of asset deteriorations were simulated in this experiment. Four levels of diversities in the asset deteriorations were simulated, where the final level corresponded to a case where all the assets comprising the fleet were identical.

It is believed that the similarities in the asset deteriorations manifest in the times-to-failures observed within the fleet, which are the only inputs needed to obtain the empirical log-hazard plots. In this experiment, the times-to-failures of the assets belonged to one of the five zones, corresponding to $100, 200, 300, 400, 500$ time-steps of operations. These zones further comprised of four points each, separated by a gap of 20 time-steps with the corresponding zone at centre. For example, the zone at 100 comprised of points $70, 90, 110, 130$. As such, there are total 20 points across the timeline. An asset's time-to-failure was simulated using a Gaussian with one of these points as its mean and a variance of 15.

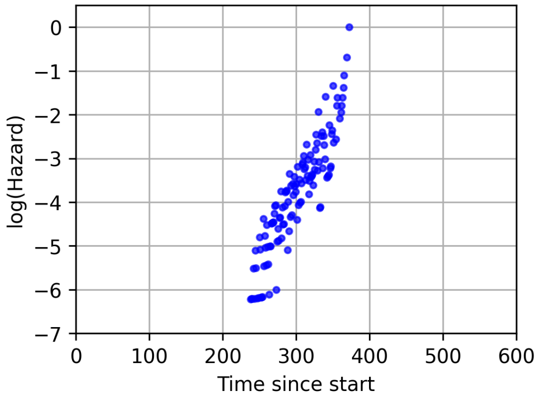
A fleet corresponding to high level of diversity would comprise of assets showing a wide range of times-to-failures. To simulate the times-to-failures in the fleet characterised by high diversity of the asset deteriorations, the assets were randomly allocated one of the 20 points and the above described method was used to simulate their times-to-failures. For the fleet corresponding to medium diversity, the assets were randomly allocated a point but this time belonging to the same zone (300) and the above method was used to simulate their times-to-failures. For the fleet with low diversity, all the assets belonged to the the same point (300), and their times-to-failures were simulated using a Gaussian with mean 300 and variance 15. Lastly, the fleet comprising of assets with identical deterioration, where all the assets had the same initial health and also the same rate of degradation, the assets had the same time-to-failure.

The empirical log-hazard curves obtained for each of these clusters are presented in Figure 4.9, with the corresponding clustering accuracy mentioned in the sub-captions. It should be noted that a high diversity in the asset deteriorations are represented by a wider range in the empirical log-hazard plot for that fleet. Moreover, the log-hazard plot of such a fleet is split into multiple functions. This means that the efficiency of the clustering algorithm, for identifying the sub-fleets of similarly deteriorating assets, manifests as the reduction in the range of the empirical log-hazard curves of the resulting asset clusters and also them not splitting into multiple functions.

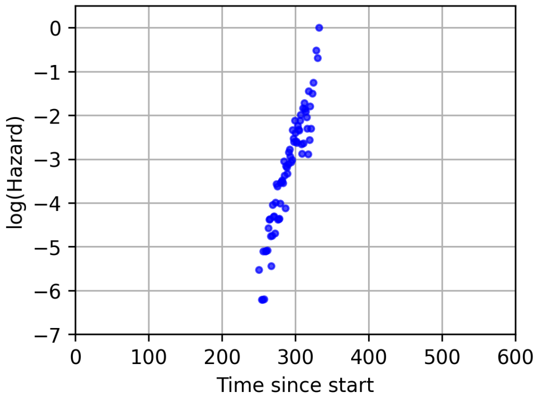
Comparing with the log-hazard plots obtained for the C-MAPSS dataset, it can be observed that the fleet-wide log-hazard curve presented in Figure 4.4 shows a wide range and is split into multiple functions. However, after clustering the log hazard plots become narrower which proves that the clustering procedure presented in Section 3 enables identifying the sub-fleets of similarly deteriorating turbofans.



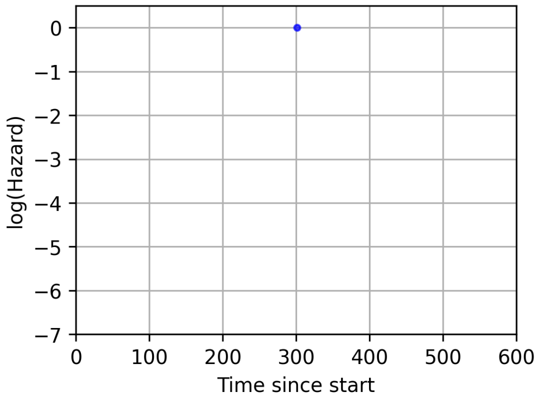
(a) Fleet 1: High diversity



(b) Fleet 2: Medium diversity



(c) Fleet 3: Low diversity



(d) Fleet 4: Identical assets

Fig. 4.9 The effect of diversity in the asset deteriorations on the empirical log-hazard curves.

4.5 Fleet-wide and Cluster-specific Weibull Density Functions

This section demonstrates the problem of high bias and high variance, associated with estimating the fleet-wide and independent cluster-specific estimates of Weibull density functions respectively.

The times to failures observed for a fleet of machines are often modelled in reliability engineering using a Weibull distribution, due to its versatility. Consider a fleet comprising of $K \in \mathbb{Z}+$ asset clusters indexed as $\{1, 2, ..., k\}$, each comprising of $I_K \in \mathbb{Z}+$ assets indexed as $\{1, 2, ..., i\}$. The probability of failure $f(t_{(i,k)}, \alpha_k, \beta_k, \gamma_k,)$ at time t for the i^{th} asset in k^{th}

cluster is given according to a 3-parameter Weibull distribution as:

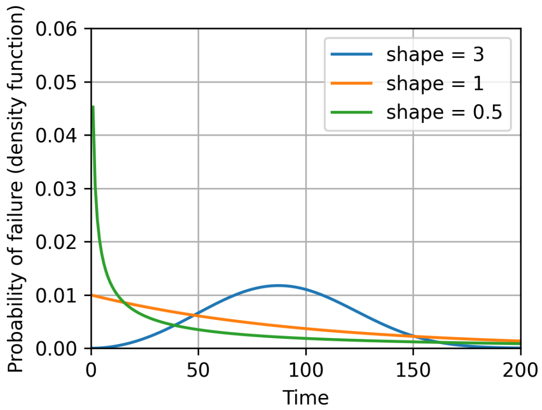
$$f(t_{(i,k)}, \alpha_k, \beta_k, \gamma_k,) = \frac{\alpha_k}{\beta_k}\left(\frac{t-\gamma_k}{\beta}\right)^{\alpha_k - 1} e^{-\left(\frac{t-\gamma_k}{\beta_k}\right)^{\alpha_k}} \tag{4.2}$$

Where α, β, and γ are the shape, scale, and location parameters of the Weibull distribution.

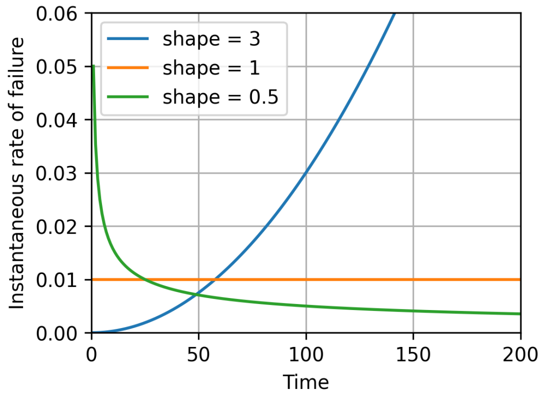
The shape parameter α plays an essential role in planning the maintenance activities. Weibull distributions with $\alpha < 1$ indicates a decreasing hazard with time, which is also known as infantile failure rate as it mostly occurs in the early life of a machine due to manufacturing defects. Similarly, constant and increasing hazard are indicated by $\alpha = 1$ and $\alpha > 1$ respectively. The increasing failure rate is often observed towards the end of life of the assets. The effect of α values on the shape of Weibull distribution and on the failure rate is shown in Figure 4.10. Hazard rate ($\lambda(t)$), also called as the instantaneous rate of failure, for a two-parameter Weibull failure density function is formulated as:

$$\lambda(t) = \left(\frac{\alpha}{\beta}\right)\left(\frac{t}{\beta}\right)^{(\alpha-1)} \tag{4.3}$$

β represents the characteristic life of the fleet, or the point at which 63.2% assets have failed. The location parameter γ is used to shift the x-axis for assets whose RUL must be predicted from a non-zero point in time axis. The location parameter is not applicable to the C-MAPSS as the turbofans in C-MAPSS fleet all commence their operations at time 0, and undergo a single failure only. The location parameter γ is therefore set to 0 in the current analysis, and ignored.



(a) Effect of the shape parameter (α) on the Weibull density function.



(b) Effect of the shape parameter (α) on the Weibull hazard function.

Fig. 4.10 The above figures show the effect of the shape parameter (α) on the Weibull density and hazard functions, with the scale parameter (β) fixed at 100.

4.5.1 Fleet-wide Weibull Density Function

The fleet-wide model corresponds to a single Weibull density function, from which the times-to-failures observed across the fleet are sampled. This is equivalent to *complete pooling* of the data, and ignores the diversity of degradation behaviours observed across the fleet. Let $\mathbf{T}$ denote the array of times-to-failures across the fleet. In a Bayesian manner, the Weibull parameters are inferred by treating the entire fleet as a single cluster and placing a non-informative uniform prior over the Weibull parameters as:

$$\begin{aligned} \alpha &\sim \mathcal{N}(0, 1000) \\ \beta &\sim \mathcal{N}(0, 1000) \\ \mathbf{T} &\sim \mathcal{W}(\alpha, \beta) \end{aligned} \tag{4.4}$$

Where $\mathcal{W}$ represents the Weibull distribution. The parameters of Weibull are inferred using MCMC, via the no U-turn implementation of Hamiltonian Monte Carlo in the probabilistic programming language Stan [168] [1]. Figure 4.11 shows the mean and first standard deviation of the posterior of Weibull density function obtained from a fleet-wide model.

[1]The same inference technique is used throughout the results presented in this Chapter.

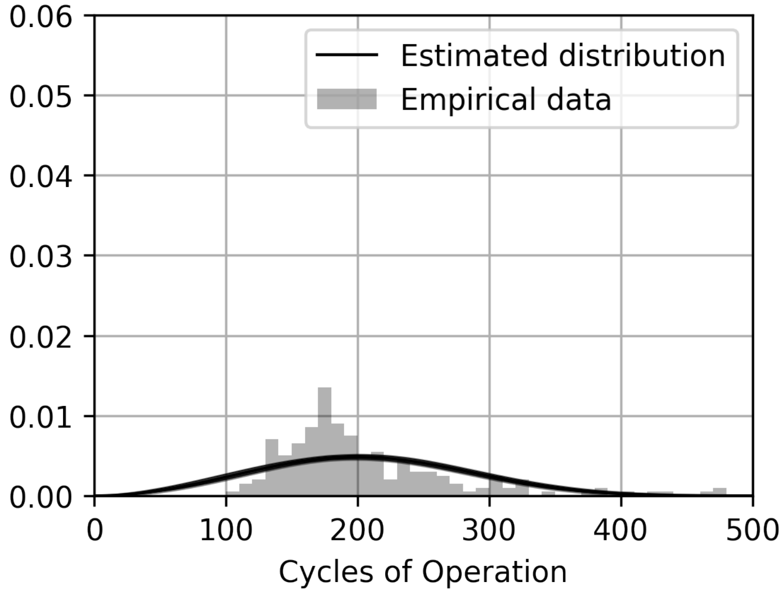
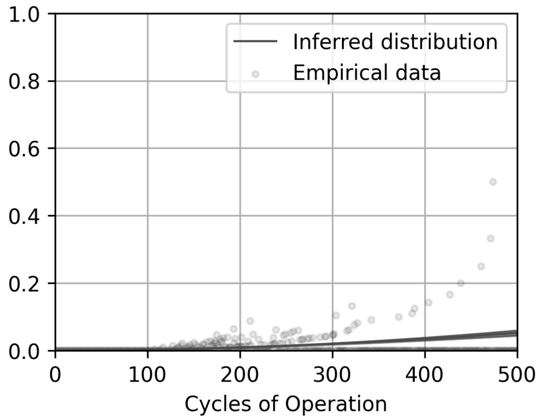
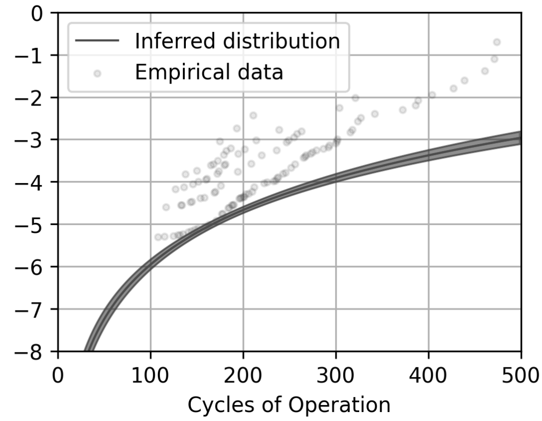


Fig. 4.11 Mean and first standard deviation of the Weibull density function posterior obtained from a fleet-wide model.

Figure 4.12 shows the mean and first standard deviation of the posterior of Weibull hazard and log-hazard functions inferred from a fleet-wide model. Log-hazard plots are shown here as they make the advantage of the hierarchical model more interpretable compared to the density function plots. Figure 4.12 shows the empirical and inferred curves for the fleet-wide data.



(a) Inferred Weibull hazard function vs the empirical data.



(b) Inferred Weibull log-hazard function vs the empirical data.

Fig. 4.12 Inferred hazard and log-hazard plots from the fleet-wide model.

4.5.2 Independent Weibull Density Functions

Similar to the fleet-wide model, the Weibull parameters are again estimated with non-informative uniform priors for each of the clusters independently. In the case of cluster-specific models however, it is believed there exists independent Weibull distributions from which the times-to-failures of all the assets in that cluster are sampled. Again, let $\mathbf{T}_k$ denote the array of times to failures observed in the cluster k. These are sampled from the corresponding independent Weibull distributions as:

$$\begin{aligned} \alpha_k &\sim \mathcal{N}(0, 1000) \\ \beta_k &\sim \mathcal{N}(0, 1000) \\ \mathbf{T_k} &\sim \mathcal{W}(\alpha_k, \beta_k) \end{aligned} \tag{4.5}$$

Figure 4.13 shows the mean and first standard deviations of the inferred posteriors for each of the cluster-specific models. The clusters are represented with separate colours and the corresponding cluster IDs are mentioned alongside the plots. A similar plot for the inferred Weibull log-hazard is shown in Figure 4.14

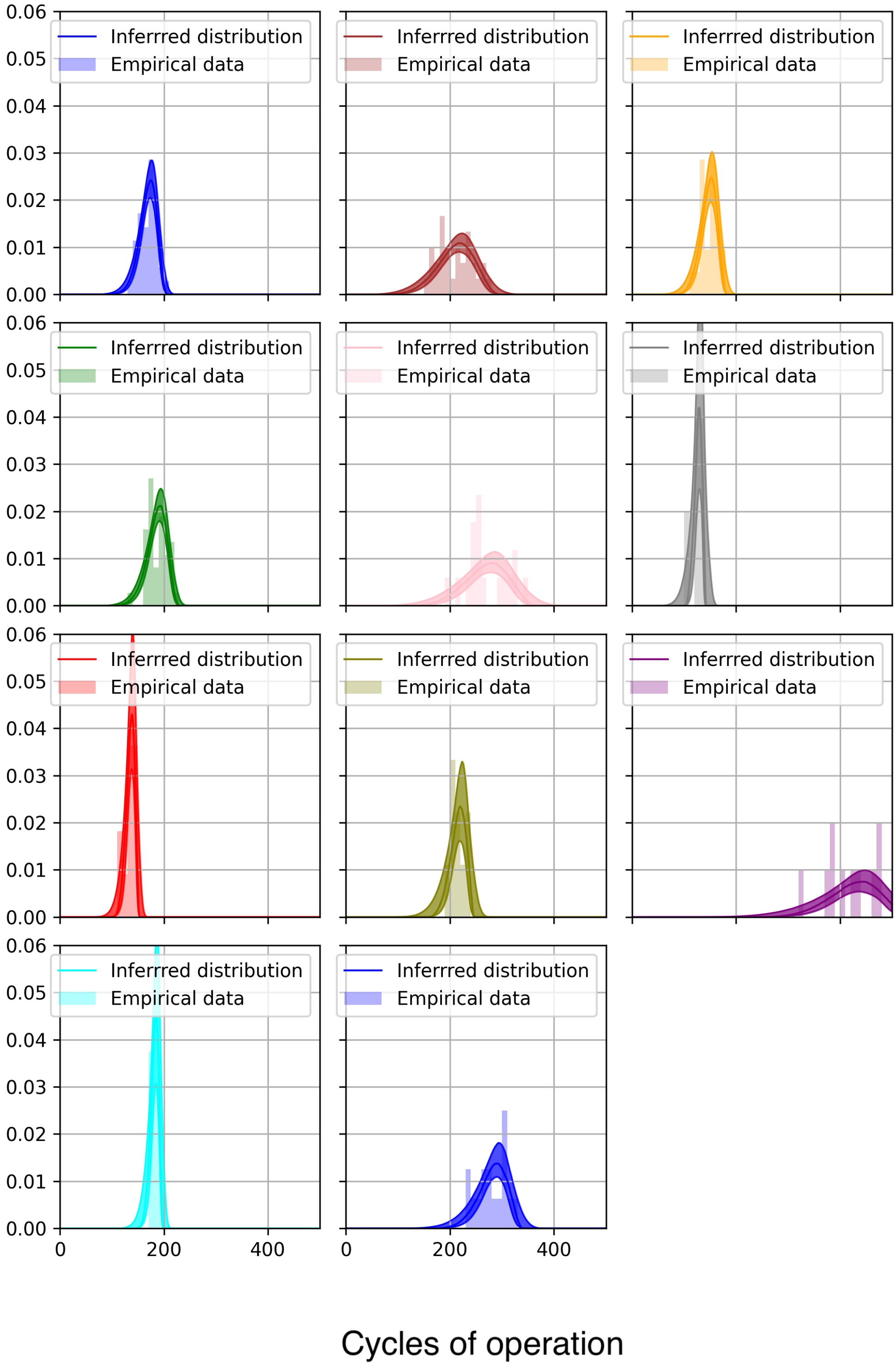


Fig. 4.13 Mean and first standard deviation of the Weibull density function posteriors inferred from the cluster-specific independent models.

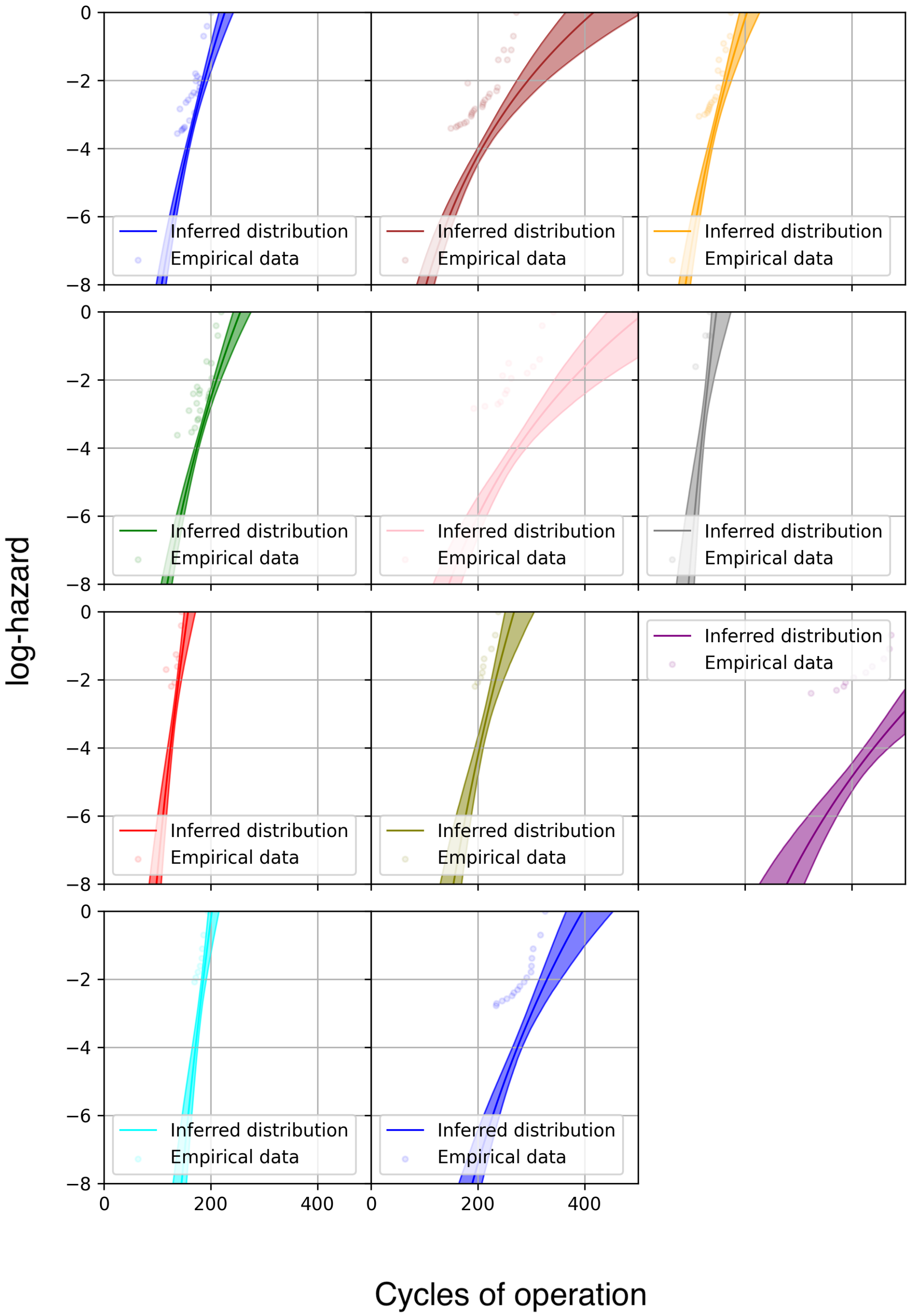


Fig. 4.14 Mean and first standard deviation of the Weibull log-hazard function posteriors inferred from the cluster-specific independent models.

4.5.3 Observations

Figures 4.11 to 4.14 show that while insufficient information for robust independent cluster-specific models, the data across the fleet are too dissimilar to be approximated by a single function for the observed failures.

1. The posteriors in Figures 4.13 and 4.14 show larger uncertainties, especially in clusters with sparse data (clusters 2 and 5). The cluster-specific models poorly approximate the Weibull density functions, as they fail to consider valuable information that can be shared across the asset clusters.

2. The posterior in Figures 4.11 and 4.12, although showing very low uncertainty, does not accurately describe the failures observed in the fleet. This is depicted by the high variance of the estimated Weibull distribution itself.

However, since the turbofans are derived from a fleet of assets operating in same conditions/ underlying failure modes, it is the Weibull distributions are similar between the clusters. The independent cluster-specific models might therefore improve by learning the parameters in a joint inference over the whole population. The following Section demonstrates using a statistical hierarchical model to learn a separate (but correlated) Weibull density function for each cluster in combined inference, where the parameters are *encouraged* to be similar.

A note on downward shift of the log-hazard curves:

The inferred log-hazard curves shown in Figures 4.12, 4.14, and 4.17 show a downward trend compared to the empirical data. This occurs because of the existence of intervals with no turbofan failures between the first and last failures. The intervals with no failures are associated with a hazard equal to 0, and therefore *pull* the hazard curve downwards.

This phenomenon is more apparent in Figure 4.12a where the inferred hazard curve is affected by the intervals associated with 0 hazard, compared to the clusters in Figures 4.17 and 4.14 with fewer intervals with no failures, such as in clusters 1, 3, 4, 6, and 7 (blue, yellow, green, grey, and red) for example. The pull towards the x-axis in Figure 4.12a is amplified while taking the logarithm, where the intervals associated with 0 hazard using the are ignored while plotting the empirical log-hazard curves (because $log(0)$ is not defined). From a reliability perspective, the signage of the Weibull shape parameter is what matters, and in all the clusters identified here there is a need of maintenance as all of them are characterised by an increasing failure rate.

4.6 Hierarchical Weibull Model for Modelling Times-to-Failures

The hierarchical model for prognosis used here is an extension of the independent cluster-specific models defined in Section 4.5.2. While similar to Section 4.5.2 where the data for each cluster are independently sampled from the corresponding Weibull distributions, a hierarchical model defines two Normal distributions at a higher level from which the Weibull parameters of the clusters are sampled. Concretely, the α_k and β_k parameters of the clusters are each sampled from their corresponding Normal distributions at a higher level. This hierarchical Weibull model, shown as a block diagram in Figure 4.15, is mathematically described as:

$$
\begin{aligned}
\mu_\alpha &\sim \mathscr{N}(0,1000)\\
\sigma_\alpha &\sim \mathscr{IG}(1,1)\\
\\
\mu_\beta &\sim \mathscr{N}(0,1000)\\
\sigma_\beta &\sim \mathscr{IG}(1,1)\\
\\
\alpha_k &\sim \mathscr{N}(\mu_\alpha,\sigma_\alpha)\\
\beta_k &\sim \mathscr{N}(\mu_\beta,\sigma_\beta)\\
\\
\mathbf{T_k} &\sim \mathscr{W}(\alpha_k,\beta_k)
\end{aligned}
\tag{4.6}
$$

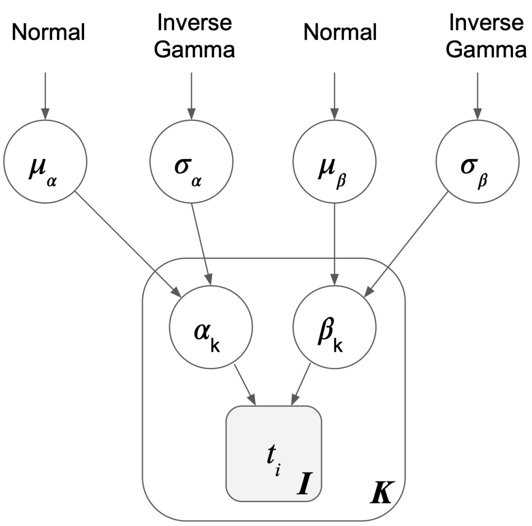


Fig. 4.15 Block diagram of the hierarchical Bayesian model.

The higher level parameters μ_α and μ_β denote the *average* behaviour of the corresponding Weibull parameters observed in the fleet. The variance parameters σ_α and σ_β govern the *extent of learning* across the clusters. A higher variance indicates that the clusters must be dissimilar, and vice versa. While the variance can be manually defined based on operational knowledge, it can also be inferred from the data. To enable being inferred from the data, the mean and variance parameters of the higher level Normal distribution are provided with their conjugate hyper-priors, i.e. Normal and Inverse-gamma distributions respectively, at a further higher level.

Figures 4.16 and 4.17 show the inferred Weibull density functions and the Weibull log-hazard functions for the corresponding clusters, obtained from the hierarchical model.

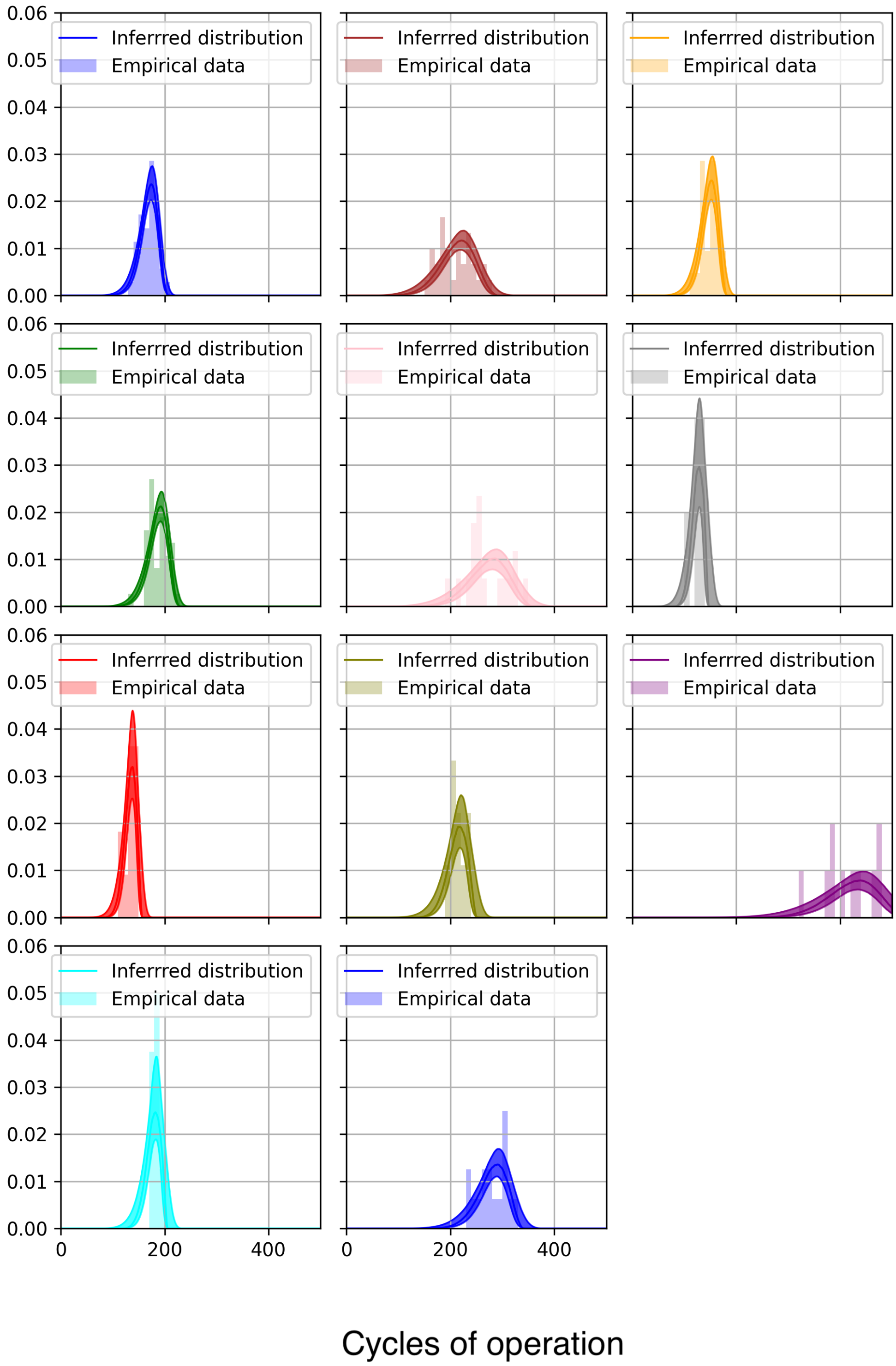


Fig. 4.16 Mean and first standard deviation of the Weibull density function posteriors inferred from the hierarchical models.

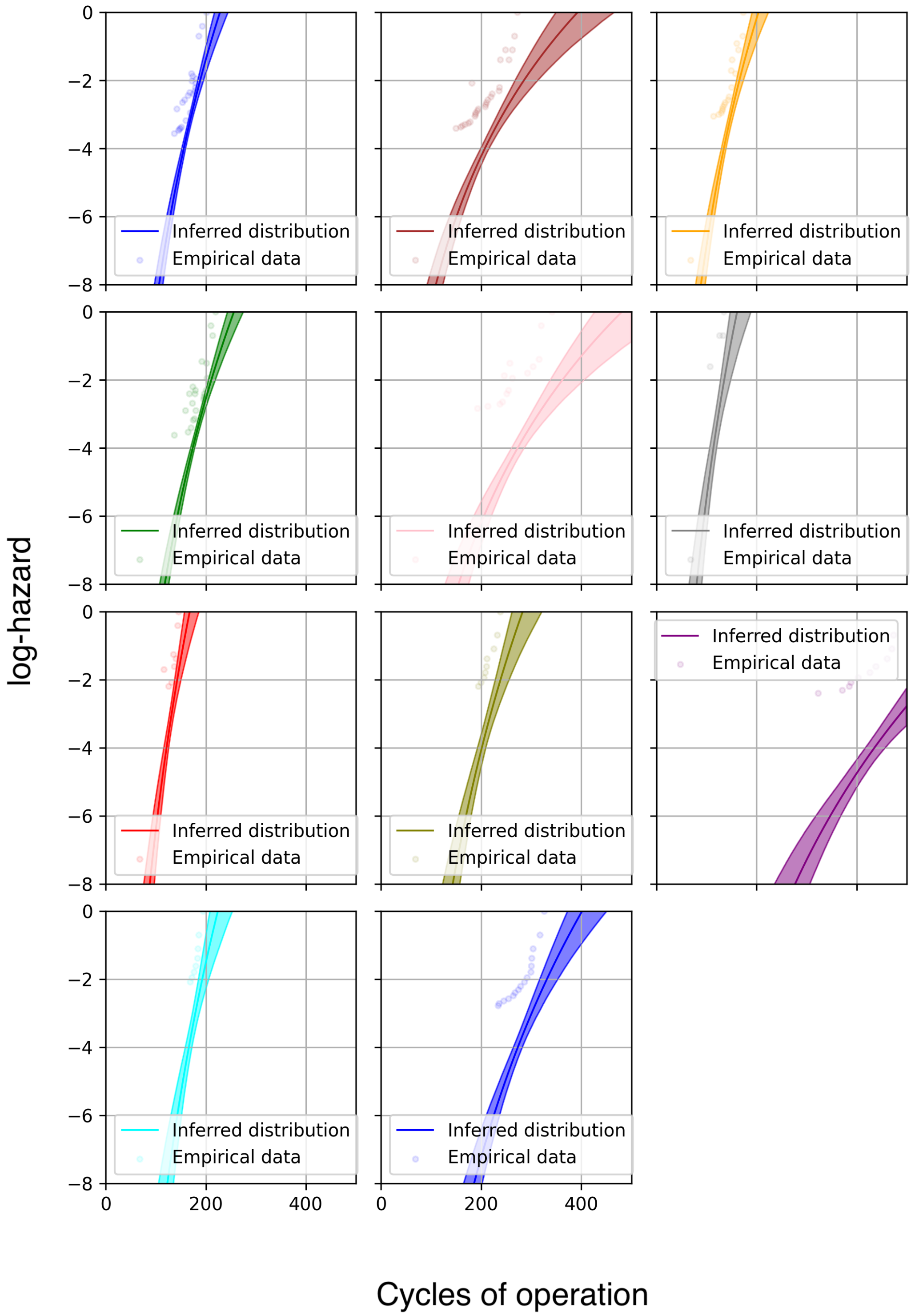


Fig. 4.17 Mean and first standard deviation of the Weibull log-hazard function posteriors inferred from the hierarchical models.

4.7 Experimental Results

This section presents the results obtained from a set of experiments conducted to analyse the efficacy of the proposed hierarchical model. Three experiments were conducted in total, each analysing the effect of the number of failures observed in the cluster, the effect of higher level parameters on the *extent of learning*, and the prediction capability of the hierarchical model for the testing dataset.

4.7.1 Hierarchical vs Independent model for Incremental Failures

The first experiment was conducted to compare the performances of the hierarchical and independent models, for the increasing number of failures observed in an asset cluster. To that end, the number of failures observed in the asset cluster 3 (selected randomly) were gradually increased from 2 until both, hierarchical and independent, models converged. The number of failures observed in the other clusters were kept constant at their original values while the failures in the cluster 3 were increased. Figure 4.18 presents the inferred values of the shape (α) parameter, which is critical for maintenance planning, as the number of observed failures in cluster 3 were increased.

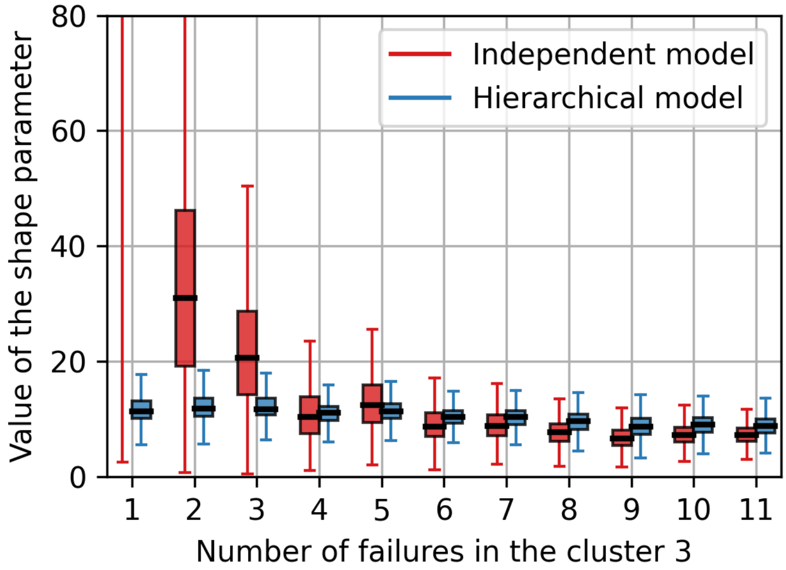


Fig. 4.18 Inferred values of α_3 vs increasing number of failures in cluster 3.

Figures 4.12, 4.14, and 4.17 show that the variance is almost equivalent throughout all the three models considered here - i.e. fleet-wide, independent cluster-specific, and hierarchical.

This is due to the fact that sufficient data is available in this fleet for all the models to converge.

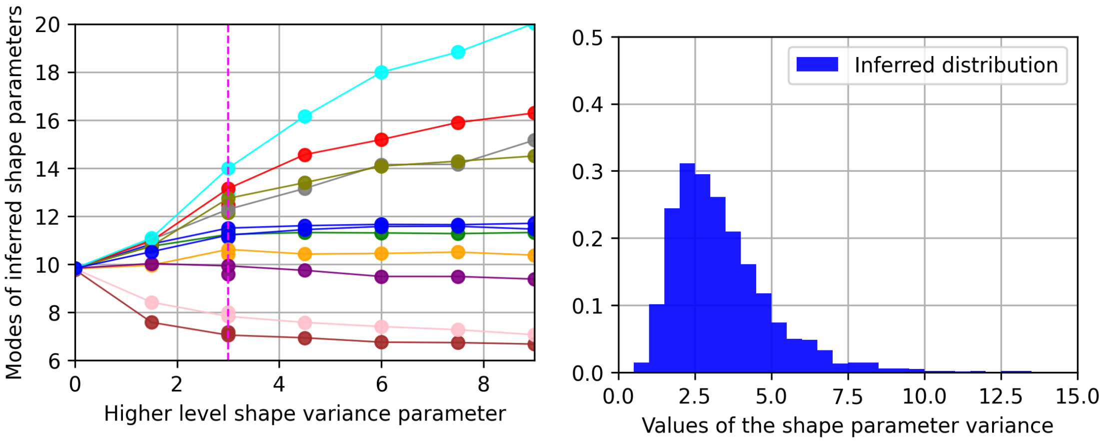
However, Figure 4.18 shows that the hierarchical model shows less variance where the data is sparse (less than 7 failures in this case) compared to the independent cluster-specific model. The median estimates of the hierarchical model are much closer to the converged value as the number of failures were increased. The hierarchical model also shows less variance in the estimations than the independent cluster-specific model.

4.7.2 Effect of Higher Level Variance Parameters

The second experiment was conducted to analyse the effect of the variance parameters (σ_α and σ_β) of the higher level Normal distributions on the corresponding inferred Weibull parameters. The hyper-priors of σ_α and σ_β were iteratively replaced with a manually set value, that varied from low to high. While the value of either σ_α or σ_β was varied, the value of the other higher level variance parameter was kept constant at the inferred value using a cluster-specific independent model. This resembles a mixed effect model, where one of the Weibull parameters is learnt hierarchically whereas the other is learnt from the cluster-specific data only. Such a mixed effect model enabled isolating the variance of only the parameter (α or β) being analysed.

The corresponding Weibull parameters estimated across the clusters for the various σ_α and σ_β values are shown in Figures 4.19 and 4.20 respectively. In the same figures, the values of the corresponding σ_α or σ_β inferred from the data (using a hyperprior) are also indicated. The modes of the inferred higher level parameters (Figures 4.19b and 4.20b), are indicated in the corresponding variance plots (Figures 4.19a and 4.20a) using vertical magenta dashed line. [2]

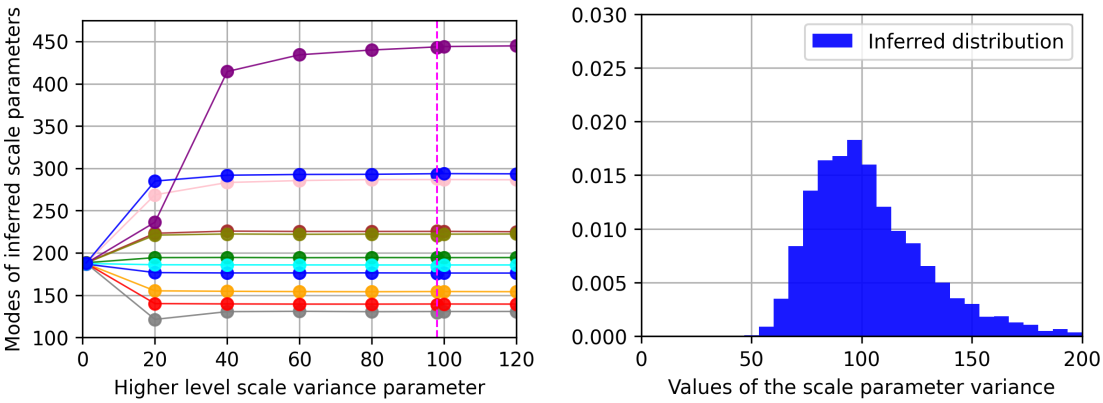
[2] This analysis, to present the effect of higher level parameters, is similar to that shown in https://github.com/omarfsosa/tech-talk-hierarchical-models for the prediction of Radon gases in the American households.



(a) Effect of σ_α on the inferred α_k for the clusters.

(b) Inferred σ_α values using a hyperprior.

Fig. 4.19 The above plots show the effect of σ_α on the *extent of learning* α_k across the clusters



(a) Effect of σ_β on the inferred β_k for the clusters.

(b) Inferred σ_β values using a hyperprior.

Fig. 4.20 The above plots show the effect of σ_β on the *extent of learning* β_k across the clusters

An added advantage of the hierarchical model is seen in Figures 4.19 and 4.20 where it (i) enables the operators to manually specify the *extent of learning* across the fleet, and (ii) provides a general high-level operating regime of the fleet. This also means that the Bayesian hierarchical model proposed herewith is transparent, unlike the Machine Learning algorithms such as neural networks. As the higher level variances keep increasing, the hierarchical model becomes equivalent to a group independent cluster-specific models.

4.8 Estimating the Remaining Useful Lives

This section describes the procedure for estimating the RULs of the operating assets. In the context of this chapter it corresponds to estimating the RULs of the 30 turbofans comprising the testing dataset.

Consider a turbofan A that has been operating for t_A times-steps. It is known here that the failure is incipient at time $t = 0$, and therefore the condition data available until time t_A is a part of its failure trajectory[3], i.e. $\mathbf{x}_n^{0:t_A}$. Conventionally, the RUL of the operating asset is estimated based on the times-to-failures of the turbofans from the training dataset deteriorating similarly until time t_A. To estimate the RUL of the operating turbofan, a Weibull distribution is often used to model the times-to-failures of the turbofans from the training dataset deteriorating similarly. The mode of the inferred maximum a-posteriori (MAP) Weibull distribution corresponds to the expected time of failure $t_A^{f_e}$, and therefore the RUL estimate for the operating asset would then equate to $(t_A^{f_e} - t_A)$.

The Weibull parameters inferred by the steps described above are bound to be associated with a high variance if the number of similarly deteriorating turbofans in the training dataset are sparse. On the other hand, the inferred Weibull distribution would be associated with a high bias if all the turbofans in the training dataset are used for estimating the RUL. The problem of sparse data is especially true as the asset approaches failure, which is described in the following paragraphs. Hierarchical model provides a systematically enables stabilised RUL predictions when using a sparse sub-fleet of similarly deteriorating assets.

In the first step to implement the hierarchical model for real-time predictions, data corresponding to the initial t_A time-steps in the failure trajectories comprising the training dataset are used to identify the sub-fleets of similarly deteriorating turbofans in the training dataset. Sub-fleets are identified using the same procedure explained in Section 4.3 and Algorithm 3 where a quadratic polynomial is fit to each condition parameter in the time-series, followed by PCA, and k-means clustering.

Hierarchical Weibull model is then used to model the times-to-failures corresponding to each of the sub-fleets, followed by identifying the sub-fleet most similar to the operating asset. The closest sub-fleet is identified based on the closest centroid, resulting from the k-means algorithm, for the observed failure trajectory of the operating turbofan. The same procedure, including the scaler and the PCA parameters, used to transform the training dataset must be used to transform the condition data from the operating asset before identifying the closest sub-fleet. Finally the RUL is estimated for the operating asset using the mode of the MAP

[3] When the failure is not incipient at $t = 0$, the time t_A is calculated from the time of failure incipience with the failure trajectory being $\mathbf{x}_n^{t_A^{f_i}:t_A}$, where $t_A^{f_i}$ represents the time at which the failure was incipient.

Weibull distribution of its closest cluster, as $(t_A^{f_e} - t_A)$ where time $t_A^{f_e}$ corresponds to the mode of the MAP Weibull distribution of the closest cluster representing the expected time of failure.

The steps involved in RUL estimations for the turbofan operating until t_A time-steps using the historical failures, referred here as the training dataset, described above are summarised in the following points:

1. Sub-fleets of similarly deteriorating turbofans in the training dataset are identified using the data-points corresponding to the first t_A time-steps of their corresponding failure trajectories.

2. The hierarchical Weibull model is used to model the times-to-failures in the sub-fleets identified above.

3. The closest cluster to the time-series of the operating asset is evaluated.

4. The RUL is estimated for the operating asset as $RUL = t_A^{f_e} - t_A$, where $t_A^{f_e}$ corresponds to the mode of the Weibull distribution inferred for the closest cluster.

Figure 4.21 schematically presents the steps explained above, with a single condition parameter shown along the y-axis as an example. The partial trajectory of the operating asset is shown in thick red and the trajectories corresponding to the observed failures in black. The time-series extending beyond t_A are faded as these are not used for clustering the trajectories. Four sub-fleets are present in the observed failures in this case, which are identified in the second step, shown in the figure with different colours. The steps include modelling the times-to-failures, identifying the closest sub-fleet, and calculating the RUL which are indicated in Figure 4.21.

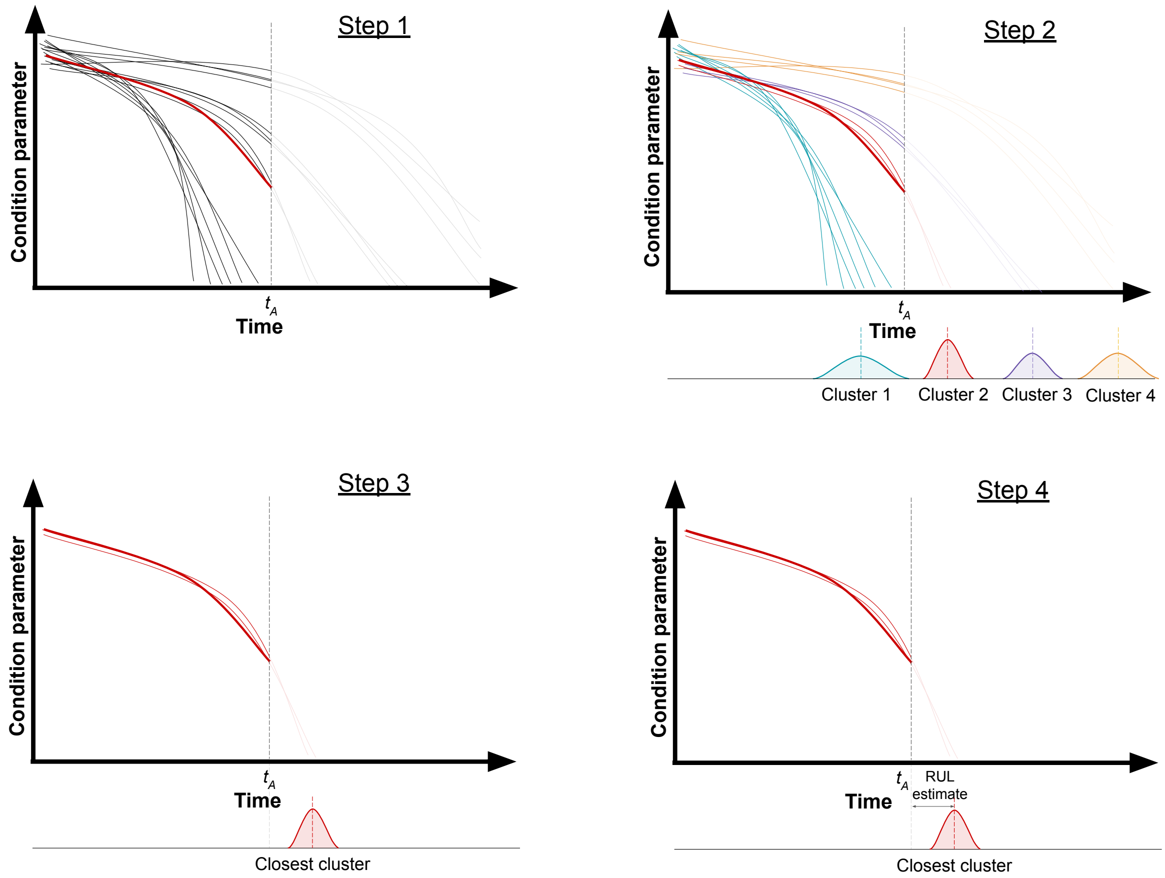


Fig. 4.21 Schematic representation of the steps involved for collaborative prognosis using the hierarchical Weibull model.

The steps shown in Figure 4.21 are repeated for RUL estimations throughout the life of the operating asset, as shown schematically in Figure 4.22. It is shown in Figure 4.22 that the sub-fleets become better defined as data is obtained from the operating asset, making the RUL predictions more accurate and confident (represented by the variance of the MAP Weibull distribution) as the failure approaches. Sub-fleets are not well defined using the data corresponding to the initial time-steps as the turbofans commence their operations in the same environment. However, with time the separation between the turbofans undergoing different rates of deteriorations becomes apparent.

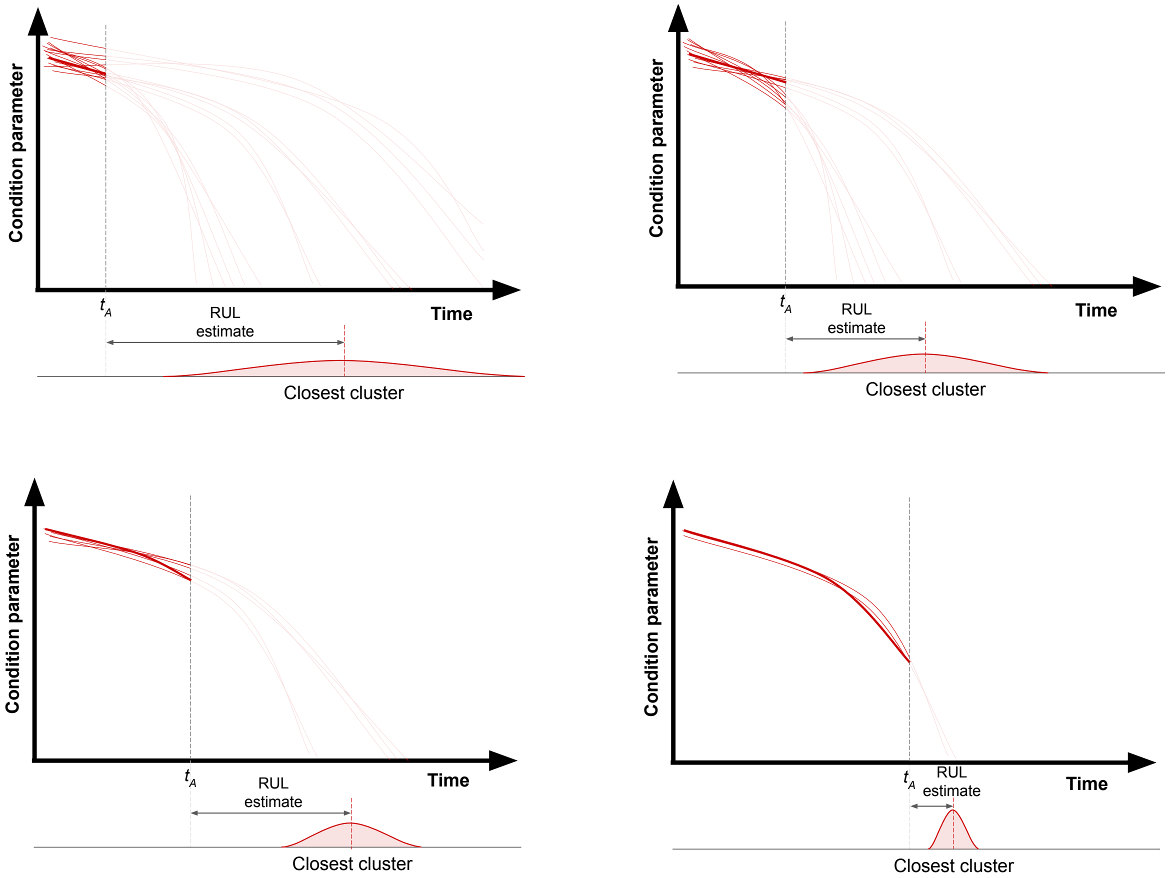


Fig. 4.22 Schematic representation of the RUL predictions using the hierarchical Weibull model along the life-time of an operating asset.

The representation in Figure 4.22 is shown for a turbofan from the testing dataset used herewith in Figure 4.23. Predictions are made at different times along the failure trajectory, shown by the corresponding MAP Weibull distribution of the closest cluster. The corresponding operating time of the asset, the expected failure time, and the actual failure time are also shown. The predictions become accurate and confident as the asset approaches its failure, both in terms of the accuracy and the confidence of the predictions.

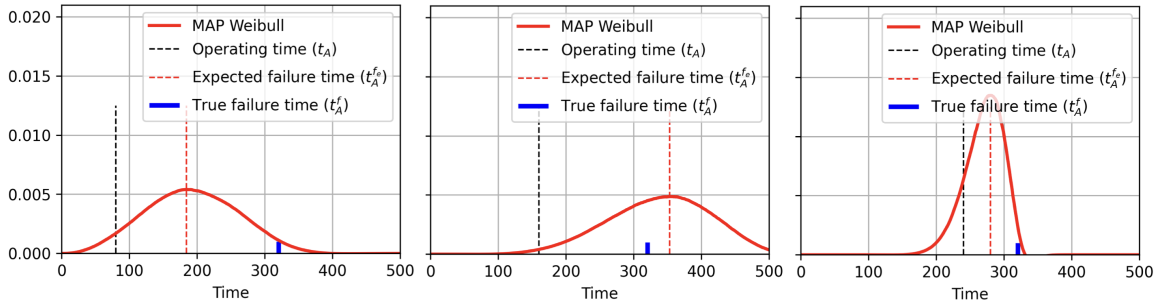


Fig. 4.23 RUL predictions using the hierarchical Weibull model, along the life-time of a turbofan from the testing dataset.

4.9 Experiments

This section describes the experiments conducted to evaluate the performance of the hierarchical Weibull model for collaborative prognosis, and also presents the results from the corresponding experiments. The discussions of the results presented in this section are presented in the following Section 4.10.

4.9.1 Predicting Failures in Simulated Turbofan Operations

The first experiment aimed at evaluating the prognosis performance by calculating the absolute differences in the actual and the predicted RULs throughout the turbofan operations comprising the testing dataset. Given the varying lengths of the failure trajectories comprising the testing dataset, the error of prediction was calculated after every 10% segment of the corresponding trajectories so that the errors can be evenly measured across the testing dataset. This evaluation is presented schematically in Figure 4.24 where two trajectories of different lengths are shown along with the points at which the differences between the predicted and estimated RULs are evaluated.

The absolute differences in the actual and the predicted RULs evaluated at every 10% segments for the turbofans comprising the testing dataset are shown in Figure 4.25. For this experiment, 15 PCA components were chosen during the clustering step that represented more than 99.5% variance in the dataset. For clustering, the threshold for reduction in the inertia[4] for adding an extra cluster was kept at 0.1, which meant that adding another cluster for the k-means algorithm would result in an overall reduction in inertia by less than 10%.

[4]inertia is the average absolute distance of the points from their cluster-centroids

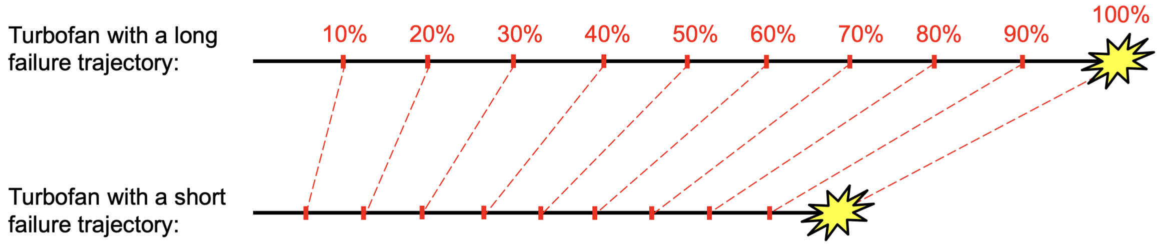


Fig. 4.24 A schematic representation of a short and a long failure trajectory, where the points after every 10% segment are marked.

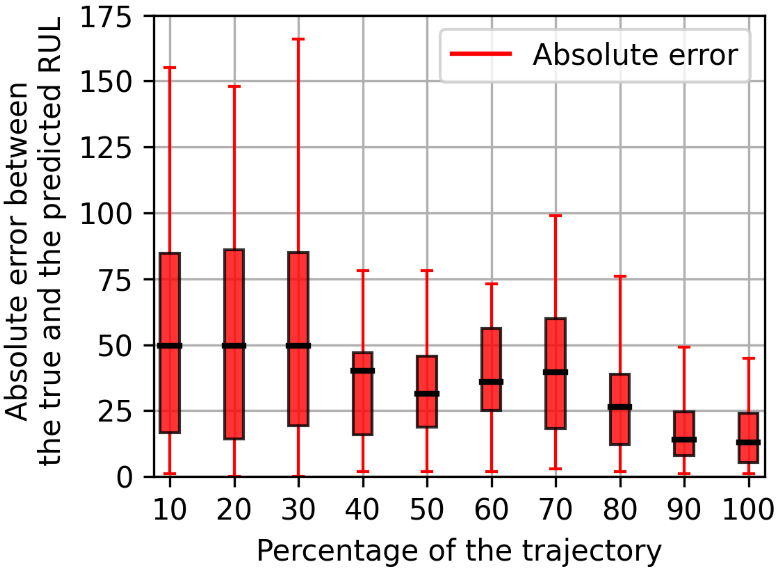
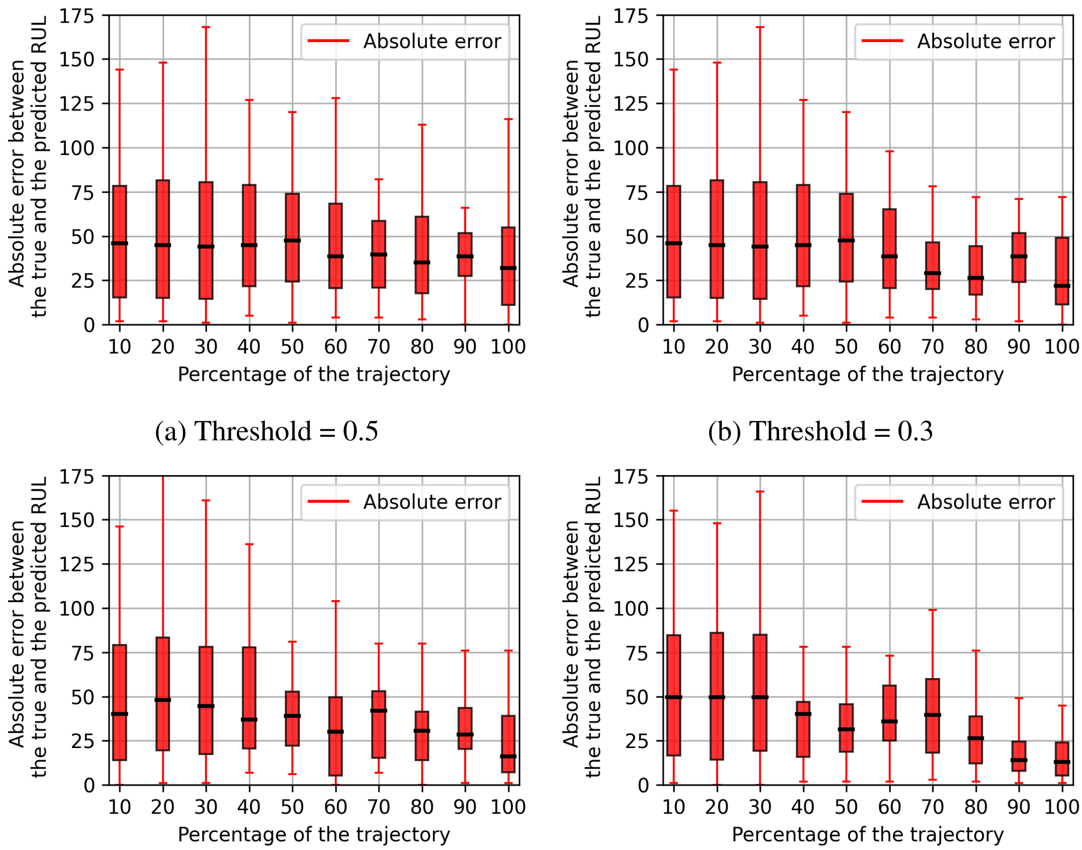


Fig. 4.25 Box-plot presenting the absolute differences in the actual and the predicted RULs evaluated at every 10% segments for the turbofans comprising the testing dataset.

4.9.2 Analysing the Effect of Clustering on the Prognosis Performance

The second experiment is an extension of the first experiment, and aims at evaluating the effect of the threshold for clustering on the prognosis performance. As explained in Section 4.3, the clustering threshold is used to determine the optimal number of clusters present in the fleet while using the k-means algorithm. The optimal number of clusters for the k-means algorithm are determined by sequentially increasing the assumed number of clusters in the fleet, until adding another cluster results in percentage reduction in the inertia below the threshold. In the experiment presented in Section 4.9.1, the threshold for clustering was set at 0.1.

In this experiment the clustering threshold is varied across a range of values at 0.5, 0.3, 0.2, and 0.1. Setting a high threshold results in fewer number of clusters in the fleet. Mathematically, a higher threshold results in higher intra-cluster distances and lower inter-cluster distances. In the current experiment, this results in a wider range of deteriorations and therefore a wider range of times-to-failures within the clusters. Figure 4.26 presents the errors evaluated at every 10% segment of the failure trajectories in the testing dataset, similar to that in Section 4.26, with a different clustering in every case.



(a) Threshold = 0.5 (b) Threshold = 0.3

(c) Threshold = 0.2 (d) Threshold = 0.1

Fig. 4.26 Errors in the predictions corresponding to various clustering threshold, mentioned in the sub-captions. A higher threshold corresponds to a wider range of times-to-failures observed within the clusters.

4.10 Conclusions from the Experimental Results

A statistical hierarchical model is proposed in this chapter to model the times-to-failures observed in an asset fleet. It is shown in this chapter that statistical hierarchical modelling inherently encourages collaborative learning for the clusters with sparse data, as it defines common prior distributions for the cluster-specific Weibull parameters at a higher level. The hierarchical model is able to mitigate the problem of high variance in cluster-specific independent models by incorporating prior knowledge about the times-to-failures from other clusters comprising the fleet. Furthermore, hyper-priors of the higher level parameters also automate the *extent of learning* across the asset clusters and enable manual intervention when needed.

It is shown in this chapter, specifically in Figure 4.18, that the technique of using hierarchical modelling for collaborative prognosis addresses the issue of high variance observed while predicting failures in sparse data clusters. It can be observed in Figure 4.18 that the hierarchical model outperforms the independent model both in terms of accuracy and variance of the estimates. It is therefore advantageous for the clusters with fewer observed failures where the independent model fails to even converge to a solution, when a cluster has fewer than 4 observed failures in this case for example. This is synonymous with the conclusions of Chapter 3, where the hierarchical model outperformed the independent model when an asset did not have sufficient data for convergence. Such clusters are ever more prevalent in modern industrial fleets due to the operators resorting to custom configurations and diverse operating conditions.

Moreover, the hierarchical model provides an added advantage to the fleet operators via the higher level distributions. A hierarchical model systematically resembles the hierarchy of information in the fleet, and therefore the distributions of the inferred parameters resemble the state of the fleet at the corresponding levels. Operators can incorporate the expert knowledge about the similarities across the fleet via the higher level parameters. This is clear from Figures 4.19 and 4.20 where the hyper-priors corresponding to the higher level variance parameters allow the model to learn the right balance of the extent of learning from the observed data. The similarities among the clusters can also be perceived from the Figure 4.19, based on the closeness of the inferred parameter values.

The procedure to implement the hierarchical Weibull model for collaborative prognosis was also described in this chapter, and demonstrated for a fleet of simulated turbofans. The conclusions derived from the results presented in Section 4.9 are summarised in the following points:

1. The length of the failure trajectories in the testing dataset range from 214 to 348 time-steps. The interquartile range of the errors of predictions while using the hierarchical Weibull model has been lower than 25 time-steps in the last two segments, and lower than 50 time-steps in the last three segments as it can be observed in Figure 4.25. The prognosis performance using the hierarchical Weibull model can therefore be deemed comparable to using the recurrent neural-networks with FedAvg, shown in [34, 33] for the case of a single failure mode. But unlike FedAvg, the hierarchical Weibull model is able to retain the prognosis performance even for the case of dual failure modes like shown in this chapter.

2. The fact that the clusters of similarly operating assets cannot be clearly identified in the early time-steps of operations is confirmed by the errors shown in Figure 4.25. In Figure 4.25 it is observed that the errors in predictions, in terms of accuracy and the variance across the testing dataset, significantly reduce as the turbofans approach the failures. The errors in predictions in the early time-steps of operations can be mitigated if the clustering is supported by other sources of information such as expert knowledge or specifications data.

3. The effect of clustering is highlighted by the results presented in Figure 4.26. In Figure 4.26, the errors in the predictions are similar across all the clustering thresholds in the early segments of the failure trajectories. In these segments, all the thresholds are equally incapable of identifying the clusters in the training dataset. However, in the later segments of the trajectories, the errors corresponding to lower clustering thresholds are significantly lower than those corresponding to higher thresholds. Comparing the errors corresponding to the thresholds 0.5 and 0.1, the errors in the final segment in the case of threshold 0.1 are one-fourth of those in corresponding segment of threshold 0.5. This is because a higher threshold results in the clusters comprising of larger number of turbofans and exhibiting a wider range of the times-to-failures, corresponding to a case of poorly clustered assets. The performance of the hierarchical Weibull model for collaborative prognosis therefore critically depends on how well the clusters are defined.

Chapter 5

Industrial Case Study: Modelling Failures in a Fleet of Heavy-duty Trucks

This chapter presents a case study for modelling the times-to-failures observed in a real-world industrial fleet using the hierarchical Weibull model presented in Chapter 4, addressing research question 2 and research objective 6 outlined in Chapter 1. The case study presented herewith is aimed at modelling the turbocharger and the alternator times-to-failures in two separate fleets of heavy-duty trucks maintained by Scania Commercial Vehicles, Sweden.

Turbochargers and alternators are critical components in heavy-duty trucks as their failures that can render the trucks immobile. Scania wishes to reduce reactive maintenance costs; specifically the operations losses associated with late goods deliveries, re-loading, and towing tucks to workshop, in addition to the spare parts and labour cost to Scania. Moreover, Scania is facing increasing demands for operational availabilities from their customers, which necessitates the shift of their maintenance policies towards predictive maintenance.

Modelling the times-to-failures is key for Scania realising their goal towards predictive maintenance policies as it enables estimating the Remaining Useful Lives (RULs) of the trucks as shown in Chapter 4. Moreover, it is beneficial for Scania to incorporate the presence of sub-fleets, or clusters, of similarly operating trucks comprising the fleet as they offer exhaustive customisation. For example, the customers are free to choose the components such as the exhaust manifold, trailer connection, braking system, etc. used in their trucks as per their liking. Customised trucks ensure best suitability for their end use, but at the same time stratify the fleet into diverse operating conditions. Discussed later in Section 5.3, most clusters are not large enough to generate sufficient failure data to model them independently from the rest of the fleet. This necessitates collaborative prognosis for Scania, and in turn the use of hierarchical modelling for the observed times-to-failures.

The dataset used for this case study includes time-series of the operations data obtained from the trucks comprising both fleets. Times-series corresponding to a given truck ranges since its start of use until a failure was observed in the target component. The dataset is referred in the following text as the Scania dataset, and the hierarchical Weibull model introduced in Chapter 4 was implemented for modelling the times-to-failures in the Scania dataset. Analyses and discussion of the results obtained along the course of this case study highlight the insights and challenges faced while modelling the times-to-failures in a real-world dataset in general, and for collaborative prognosis specifically. Moreover, since the parameters describing truck health conditions were recorded as time-series of irregularly recorded histograms, techniques to preprocess the same are also analysed as a research contribution.

Section 5.1 explains the process of obtaining the operations data from the trucks and highlights the features used for predicting the turbocharger and alternator failures for this case study. Summary statistics of the dataset and preprocessing to select qualifying failure trajectories are presented in Section 5.2. Section 5.2 also provides a brief discussion of the prevalent techniques for using histogram data to train the statistical models. Three techniques were explored to identify the clusters of similarly operating and deteriorating trucks, which are described in Section 5.3. Section 5.4 presents and discusses the results obtained after modelling the failures in the identified clusters using fleet-wide, cluster-specific, and hierarchical Weibull models. The conclusions are drawn in the final Section 5.5.

5.1 Dataset Description

This section describes the Scania dataset. Brief introductions of the turbochargers and the alternators are provided, followed by description of the structure of the dataset. Identical steps were followed to obtain the data corresponding to either component failures, and the dataset description including the structure and the features also applies to both components similarly.

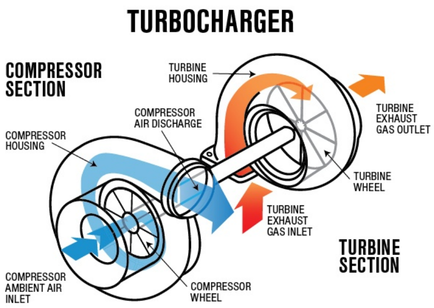
5.1.1 Targeted Components

1. **Turbocharger** is a modern improvement in internal combustion engines, that increases the density of air going into the combustion chamber of the engines. It comprises of a turbine that is driven by the engine exhaust and is connected to a compressor for engine inlet air. The exhaust drives the turbine, which is connected to the compressor that in turn increases the density of the air going into the combustion chambers.

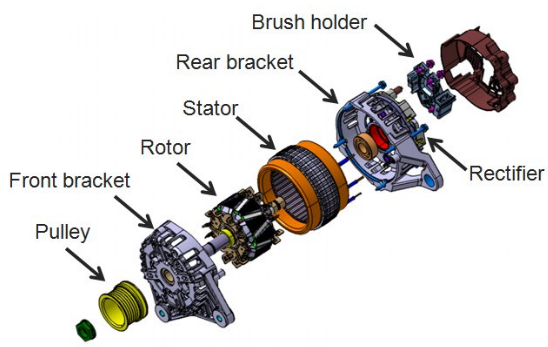
Increased density of the inlet air improves the power output and efficiency compared to a naturally aspirated engine of the same size and capacity. Further information about the turbochargers can be found in [189]. Turbochargers are critical for a truck's operations, failure of which results in unplanned vehicle downtime due to immobilisation and increased service costs for the maintenance provider.

2. **Alternator** is another critical component that converts the mechanical energy of the moving engine shaft into electrical energy (alternating or direct current depending on the use). The output of the alternator drives the electrical units on board such as the headlamps, windows, ignition, driver-support systems, etc. and also recharges the battery. Principally, alternator comprises of a rotating shaft with an electromagnetic coil, surrounded by a stator made of laminated sheets and copper wire coils. The alternator shaft is connected to the engine shaft via a belt and pulley mechanism. As the engine runs it causes the alternator shaft to rotate inside the stator and produce alternating current. Alternators in actual use however need to be further enhanced with accessories such as rectifiers and regulators. More information about the alternators can be found in [144].

Figures 5.1a and 5.1b show the turbocharger and the alternator respectively [1].



(a) Turbocharger



(b) Alternator

Fig. 5.1 Figures showing the components targeted for prognosis in this case study.

Failures commonly observed in the turbochargers include: (1) Wear WG shaft/ bushing - this is mechanical wear which always eventually happens, (2) bearing wear - which is another form of mechanical wear, and (3) Wear/ relaxation of the turbine sealing rings [147, 4, 150]. Whereas in the case of alternators, commonly observed failures include: (1) Worn out carbon

[1] These images are taken from https://www.eagleridgegm.com/what-is-a-turbocharger-and-how-does-it-work/ and [172]

brushes, (2) Stuck carbon brushes, (3) Broken components in PCB, (4) Broken diodes, (5) Worn out ball bearings, (6) Broken stator winding, or (7) Broken rotor winding [149, 179].

The data used for this case study were obtained from a fleet of trucks continuously monitored throughout a fixed period of time since the time of their production. Two separate fleets comprising of more than 30,000 trucks each were monitored for the turbocharger and the alternator failures, where just under 3000 and under 1000 failures respectively were observed in the particular study period considered for this case study. Condition data from the trucks that did not fail were used to compare the deviations with the trucks that encountered failure, but only those trucks which encountered failures in the target components in the corresponding fleets are considered for the purpose of modelling the times-to-failures. The fleets monitored for the turbocharger and the alternator failures are referred herewith as the *turbocharger fleet* and the *alternator fleet* respectively.

Data representing each truck was a time-series of datapoints, each datapoint comprising of 150 technical specifications and 37 operational parameters, ranging since the trucks were out of production until a failure was observed in the target component. The datapoints including the technical specifications, and the operational parameters in the time-series of a truck are referred herewith as *snapshots* that represent the truck's cumulative operating regime until that time. Moreover, the snapshots were only recorded when certain conditions were met which included the truck's age, data contract with the customer, network capability, workshop visits, etc. The time-series were therefore irregular and sometimes sparsely recorded, in the sense that the gap between the consecutive snapshots was not consistent across the fleet or even for a single truck.

The technical specifications are static categorical variables that describe the truck's characteristics such as model, engine type, cab type, wheel type, etc. The operational parameters on the other hand are temporally dynamic, and obtained using sensors monitoring various internal and external operating conditions of the given truck.

The operational parameters of the trucks were recorded as histograms, the bins being the intervals spanning over the range of possible values for the corresponding parameter. The values falling these bins represented the time spent by the corresponding truck in that range of values. The histogram values were therefore additively updated along the trucks' operation.

For example, the ambient temperature histogram comprised of 9 bins as $[< -30]$, $[-30,-20]$, $[-20,-10]$, $[-10,0]$, $[0,10]$, $[10,20]$, $[20,30]$, $[30,40]$, $[40 <]$. The bins represent the range of ambient temperatures in which a given truck can possibly operate. Values in the bins continuously increase and represent the time a given truck spends in that particular temperature range. When the conditions are suitable the values of the bins are recorded and represent the range of temperature the truck has been operating in until that time. Subtracting a given histogram from the previous one represents the range of temperature the truck has been operating between the consecutive snapshots. An example of varying bin values over a course of the truck's usage is shown in Figure 5.2.

Some histograms were co-recorded as matrices, which are essentially two-dimensional histograms with either feature on each of its axes. For example a matrix of axle load vs. speed is a two-dimensional histogram, which records the time the truck spends in a given load/ speed range combination. If a possible range of load is divided into 5 bins, and that of speed into 8 bins then the total bins in a matrix would be $5 \times 8 = 40$ bins. An example of such a matrix is shown in Figure 5.3. Matrices were converted into their constituent histograms by adding the elements across the axes, illustrated in Figure 5.3.

Apart from the histogram and categorical types, certain parameters were also recorded as scalars. As such, the trucks' operating conditions and health data were a time-series of sparsely recorded histograms along with some other categorical and scalar features. The combined set of all histogram, categorical, and scalar parameters constituted the data describing a given truck in the fleet, and a representative sample of it is shown in Table 5.1. The *Date_recorded* column in Table 5.1 represents modified date for the purpose of illustrating the irregularity in the recorded data. The *Failure indicator* column indicates if the given truck encountered a failure or not. For this case study, only the trucks which encountered a failure are considered. A single row of the operations parameters recorded in a snapshot is also shown schematically in Figure 5.4 using the corresponding histogram representation.

Various histogram (ambient temperature and pressure, axle loads, boost air pressure), categorical (Cab type, engine type), and scalar (Turbocharger uptime, turbocharger load) features were used for the failure prediction analyses for Scania trucks. These features were identified to be relevant for prognosis by a combination of statistical recommendations, and also from [39] where a case study for generating artificial failure data was conducted for the components targeted herewith.

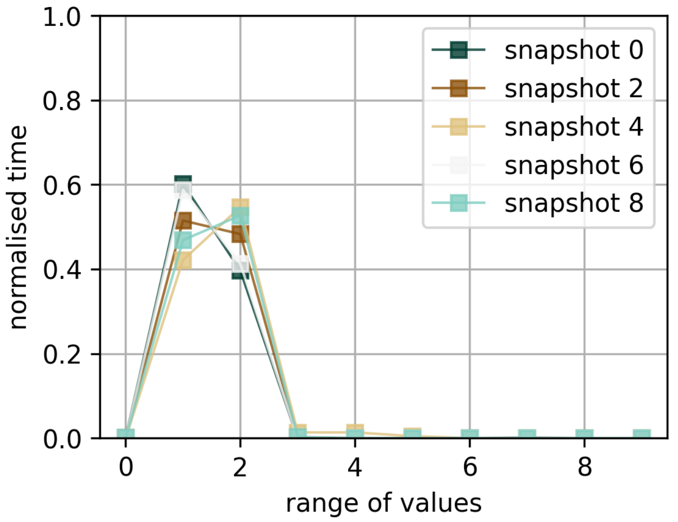


Fig. 5.2 Example of evolution of normalised histograms corresponding to the **Starter-motor time** operating parameter through a failure trajectory. X-axis represents histogram bins and the bin values are shown plotted. As the truck ages, a rightward shift from bin 1 towards bin 2 is observed in the example shown.

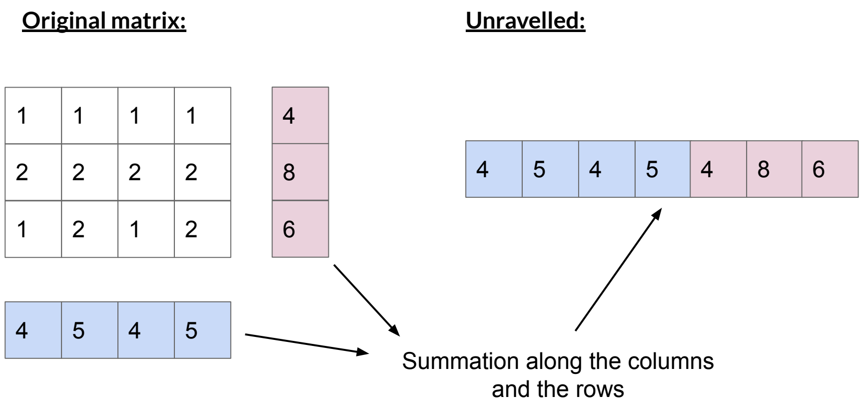


Fig. 5.3 Example of two operations parameters co-recorded as a matrix, and transforming it into the corresponding histograms.

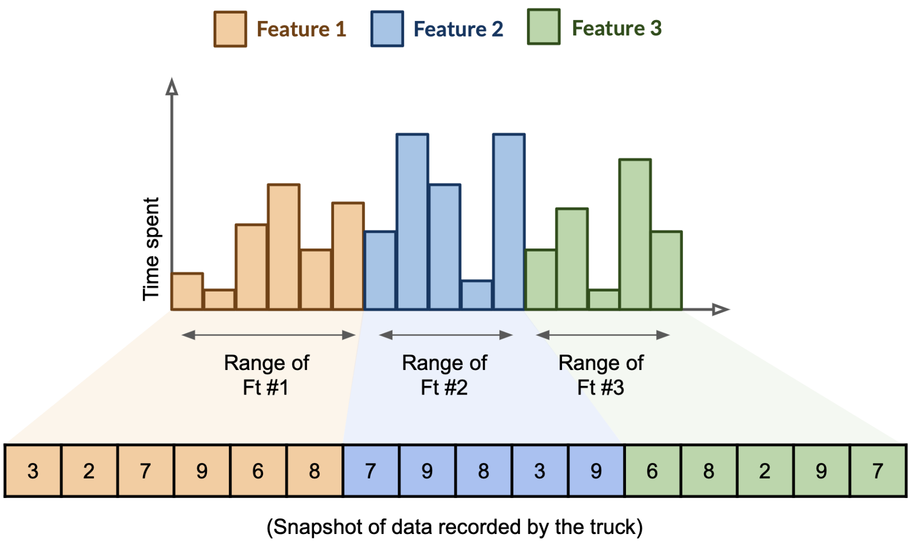


Fig. 5.4 A set of features recorded as histograms. The values in the array of recorded data correspond to the heights of the histograms, which in turn are the time spent by the truck in the corresponding bins

Table 5.1 Representative example of the Scania dataset

Truck id	**Date recorded**	**Cat. Feature**	**Scalar Feature**	**Opr Feature 1 Bin 1**	**Opr Feature 1 Bin 2**	**Opr Feature 2 Bin 1**	**Opr Feature 2 Bin 2**	**Opr Feature 2 Bin 3**	**Failure indicator**
1	01-01-2000	A	32	6	8	4	2	8	1
1	05-06-2000	A	79	14	23	13	12	34	1
1	23-11-2000	A	136	43	76	98	43	132	1
2	03-02-2000	E	14	2	9	1	6	10	0
2	16-05-2000	E	67	21	87	24	76	129	0
2	12-01-2001	E	83	54	124	36	89	154	0
2	06-07-2001	E	178	98	201	67	103	198	0
				⋮					
n	11-02-2000	A	5	2	5	1	6	10	0
n	18-06-2000	A	34	18	30	24	76	129	0
n	2-12-2000	A	85	45	86	36	89	154	0
n	24-04-2001	A	113	76	129	67	103	198	0

5.2 Preprocessing and Summary Statistics

This section presents the summary statistics of the failure trajectories in the Scania dataset including the times for which the trucks were monitored (failure trajectory length), average time between the consecutive snapshots, number of snapshots per trucks, etc. Filters applied for preprocessing the dataset, and the corresponding reasons are also discussed in this section.

Industries often record the asset health data as histograms due to their memory efficiency and homogeneity across the variables [63, 117]. Formally, the histogram data used for experiments in this case study are classified as categorical histogram data, where a frequency is assigned to each bin [37]. Histograms are different than the often expected scalar input of the ML algorithms, and must systematically be preprocessed to extract the information.

On a wider scale, complex data structures different from the commonly encountered numeric and categorical variable types are studied under the field of symbolic data analysis [37]. But to the best of authors knowledge, literature presents only few instances which specifically target prognosis using histogram data. These include the works of [45], [148], and [64, 62, 63] who investigated compressor, battery failures, and NOx sensor failures in heavy-duty trucks respectively. [45] did not clearly outline the preprocessing steps while using the histogram data, and the study of [148] was limited by the small fleet size used for analysis. The closest of the above three works to the case study discussed in this paper is that of [64, 62, 63] who used a dataset similar to the one used here, but for a different prognosis technique and target component. The authors in [64, 62, 63] used random for classifying the data corresponding to failure and non-failure class.

The degradation in the components while the trucks are idle is assumed to be negligible compared to the degradation when the components are operating. In that regard, the operating durations of the components were used to model the failures rather than the calendar days. From here on, only the actual durations of operations of the target components are used in the context of the analyses, and not the calendar days.

Moreover, the following filters were applied while choosing the failure trajectories for further analyses:

1. A filter was applied on the minimum gap between the consecutive snapshots of the operations data for a truck, because the snapshots were recorded repeatedly whenever a truck visited the maintenance depot to ensure correct measurements. The repeated measurements resulted in redundant and repeatedly identical measurements in the dataset. Moreover, ensuring a large enough gap ensured that the histograms sufficiently represented the operations profile of the trucks for the corresponding intervals. It was ensured that the consecutive snapshots were separated by at least 100 time units. For

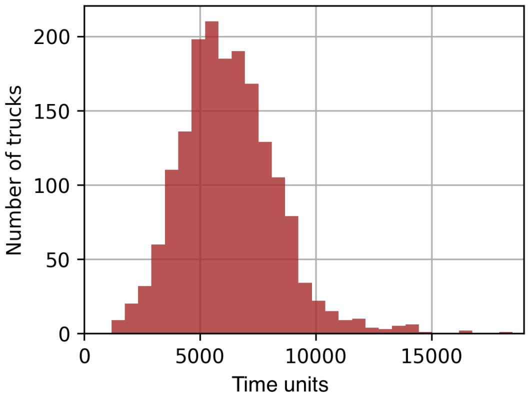
example, if the first snapshot was recorded at 0^{th} time unit, then all the snapshots recorded until 100^{th} time unit were ignored. If the next qualifying snapshot was recorded at 110^{th} time unit, then all the snapshots recorded until 210^{th} time unit were ignored, and so on.

2. Only those trucks with at least 5 snapshots, 4 of which lay in the final 2000 time units of operations, were chosen for the analyses. This ensured sufficient number of snapshots were available for clustering the trucks wherever possible.

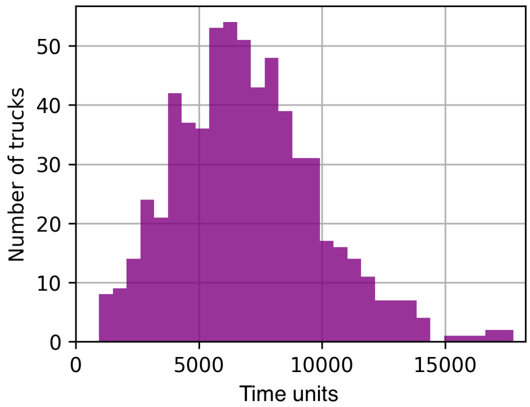
3. Lastly, filter was applied to the minimum length of the failure trajectory, of 1000 time units, to ensure that the failures occurring in the fleet are not infant mortality failures occurring due to the manufacturing defects. The times shown in the current analyses are evaluated post the infant mortality window of 1000 time units.

After the above filters were applied, total 1743 and 526 failure trajectories from the turbocharger and alternator fleets were selected respectively for further analyses.

The summary statistics of both fleets are presented in Figures 5.5 to 5.8. Figure 5.5 presents the distribution of the trajectory lengths, in time units of operations, across the turbocharger and alternator fleets. Figure 5.6 presents the distribution of number of snapshots recorded across the turbocharger and alternator fleets respectively. Figure 5.7 presents the distribution of average gap between consecutive snapshots across the turbocharger and alternator fleets respectively. This illustrates that the time-series is irregular across the trucks and also for the same truck. Figure 5.8 presents the empirical log-hazard curves corresponding to the failures observed in the turbocharger and alternator fleets. Same procedure used for the log-hazard plot in Figure 4.4b was used here for the two fleets. The idea that the Scania dataset comprises of clusters of similarly operating trucks is supported by the empirically obtained log-hazard curves, given the range of the failure trajectory lengths and that the log-hazard curve is split into several functions - as discussed in Section 4.4.1.

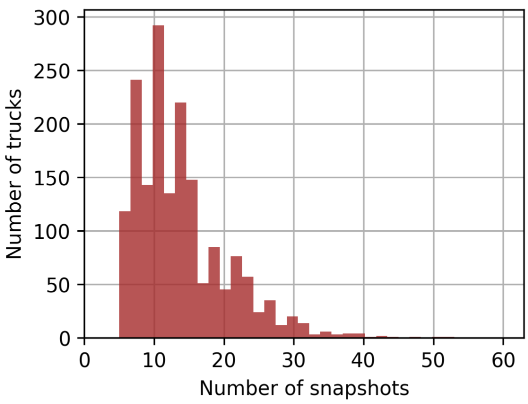


(a) Turbocharger

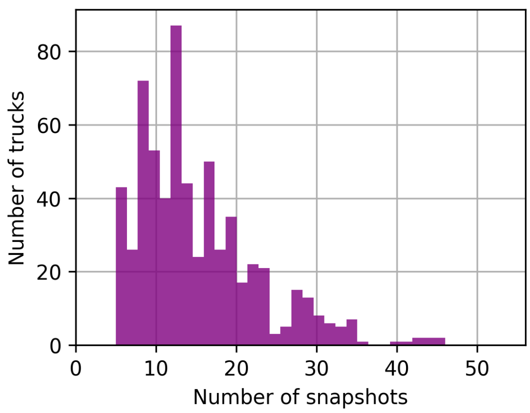


(b) Alternator.

Fig. 5.5 Histograms showing the failure trajectory lengths of the turbochargers and the alternators in time units of operations, the corresponding component mentioned in the sub-captions.

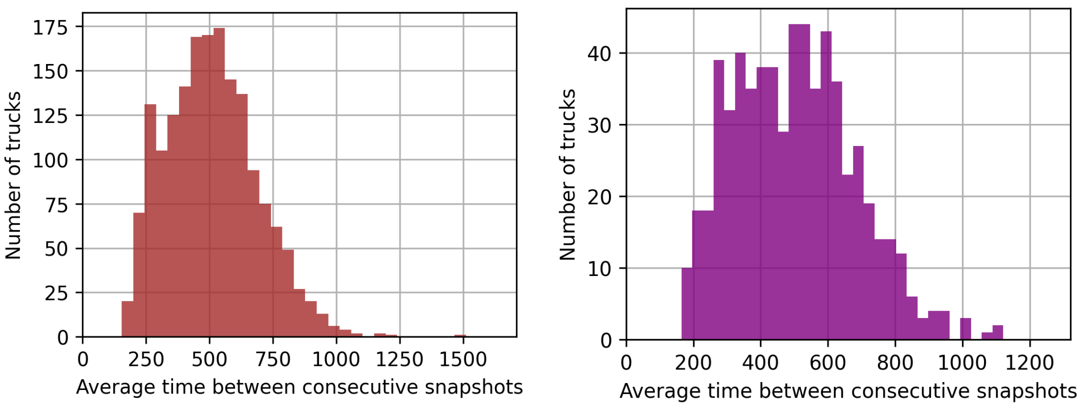


(a) Turbocharger.



(b) Alternator.

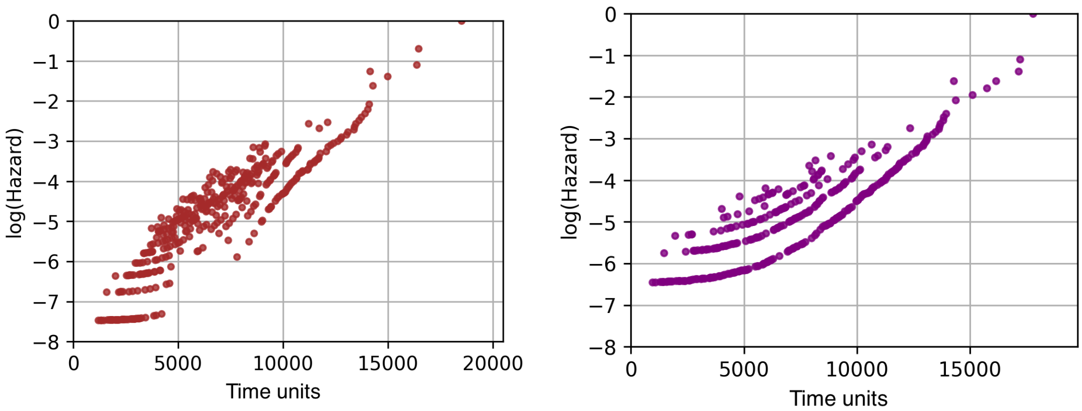
Fig. 5.6 Histograms showing the number of snapshots across the turbocharger and the alternator fleets, with the corresponding component mentioned in the sub-captions.



(a) Turbocharger.

(b) Alternator.

Fig. 5.7 Histograms showing the average durations between the turbocharger and alternator fleets in time units of operations, the corresponding component mentioned in the sub-captions.



(a) Turbocharger.

(b) Alternator.

Fig. 5.8 Empirical log-hazard curves obtained from the turbocharger and alternator fleets.

The idea that customisation of the trucks and the diversity of their operating environments leads to presence of clusters of similarly operating trucks is supported by the summary statistics of the fleet presented in this section. The failure trajectory lengths across the trucks in both fleets observed in Figure 5.5 range from 1000 till more than 15000 time units. Such range is significantly diverse and necessitates using a hierarchical model. The empirical log-hazard curves shown in Figure 5.8 further confirm this fact as for both turbocharger and alternator fleets, the log-hazard plots are split into several functions and span across a wide range of operating time units. Comparing with the experiment discussed in Section 4.4.1, this corresponds to the case where the simulated fleet comprised of clusters but no clustering was implemented while plotting the log-hazard curves.

5.3 Clustering the Scania Dataset

This section discusses three techniques for clustering the failure trajectories corresponding to either component failures in the Scania dataset. The clustering efficiencies of the techniques are also compared, based on the empirical log-hazard curves plotted for the clusters obtained in each of the cases.

It should be noted that the collaborative anomaly detection technique presented in Chapter 3 is not implemented here in the case study to identify the failure trajectories. This is due to (1) The asset condition data used for this case study were sparsely recorded in time domain, and (2) the industry experts suggested that the heavy-duty trucks used for the case study did not have any maintenance interventions until the failures, and also that they undergo steady deterioration throughout the operations. Given this, it is assumed that anomaly detection is not necessary and the beginning of the trucks' lifecycle can be treated as the start of their corresponding failure trajectories.

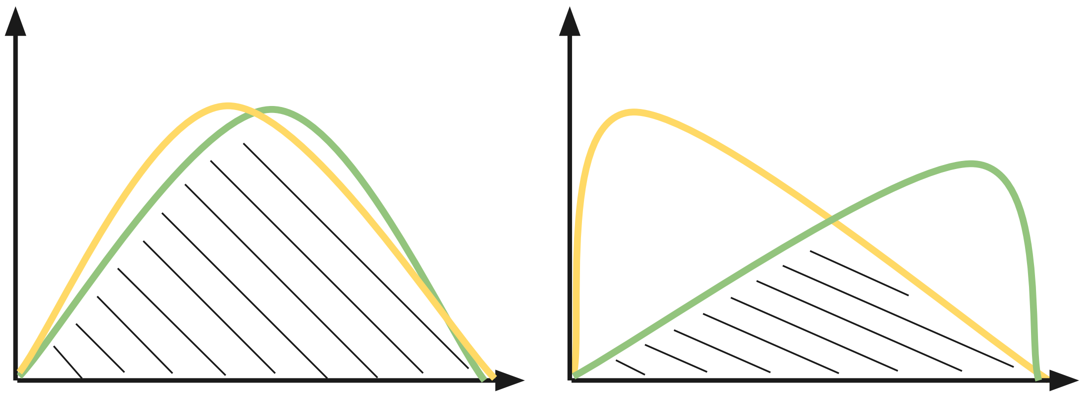
Observations noted in the experiment presented in Section 4.4.1 about the effect of the quality of clustering on the empirical log-hazard curves are used as basis for comparing the clustering techniques. It was shown in Section 4.4.1 that the log-hazard curves obtained empirically from the clusters tend to become compact as the clustering is improved, both in terms of the range of times-to-failures observed in the clusters and also by not being split into multiple functions. As such, the technique herewith that showed maximum reduction in the range of times-to-failures within a cluster was deemed suitable. The log-hazard curves obtained without any clustering for the turbocharger and alternator failures are shown in Figure 5.8 for reference of the reader.

Identifying the clusters of similarly deteriorating trucks in the Scania dataset was challenging particularly due to their operating parameters being recorded as histograms. Specific challenges while using the histogram data for clustering the failure trajectories include computational complexity while evaluating the distances, multi-modality of certain histograms, and irregularities in the time-series.

Given the challenges, the first step in two of the following techniques were aimed at transforming the time-series of histograms into corresponding point values for ease of analyses. The transformed time-series of point values were subsequently compressed for comparing across the trucks comprising the Scania dataset using the clustering technique introduced in Section 4.3, where a polynomial function is estimated for the given time-series followed by PCA and k-means clustering of the polynomial coefficients. The third technique involves using specifications data to identify the clusters of similarly operating/ deteriorating trucks in the Scania fleet.

5.3.1 Using the Intersection Areas Across the Histograms

The aim of this technique is to quantify the shifts observed in the operating regimes, represented by the histograms, of the trucks along their failure trajectories as point values. Underlying idea used to achieve this is to evaluate the similarities between two histograms in the form of their percentage intersections [141]. For example, similarity of two distributions with equal areas under the curves can be quantified by the proportion of their overlapping area. If the overlap is 100% then the distributions would be identical and vice versa. This is demonstrated in Figure 5.9 where the cases of high and low overlaps are shown. Since the histograms in Scania dataset have same number of bins across all trucks, normalising their values and evaluating overlapping areas should represent similarities between two given histograms.



(a) High overlap. (b) Low overlap.

Fig. 5.9 An example of high and low overlapping areas in two distributions with equal area under the curves. A high overlap indicates that the distributions are similar and vice versa.

The first step for implementing the above involves calculating differences across the bin values of the consecutive histograms in the time-series (i.e. $\sum_{i=1}^{n}[H_i^t - H_i^{t-1}]$), and normalising them (i.e. $H_i^t = \frac{H_i^t - H_{max}^t}{H_{max}^t - H_{min}^t}$). Here H_i^t is the i^{th} bin of the histogram consisting of n bins and recorded at time-step t, and H_{max}^t and H_{min}^t are the maximum and minimum bin values recorded for the histogram respectively. The differences between the consecutive histograms represent the operating regimes of the trucks between the time intervals when the corresponding histograms were recorded, and normalising them ensures that the histograms had the same sum of bin values.

As the trucks were monitored since production, it is assumed they started operating in a healthy condition, and that the first histogram represents the *healthy* operating regime of the given truck. Evaluating the overlap of a later recorded histogram with the first recorded histogram therefore represents the similarity of the corresponding operating regime with a healthy regime. Moreover, the immediate trends in the truck's operations were quantified by evaluating the overlaps between the consecutive histograms along its failure trajectory. The area overlap across the consecutive histograms serves as a measure for deviation in the truck's condition compared to the previous snapshot, and therefore also immediate change in its health compared to the initial condition. Evaluating the overlaps of a given histogram with the initial and the previous histogram in the time-series also ensured that the directionality was considered while reducing the histograms to point values.

To that end, each histogram was reduced to two values being: (1) its overlap with the first recorded histogram representing the truck's healthiest state, and (2) its overlap with the antecedent histogram. The overlaps were evaluated by adding the minimum among the corresponding bin values of the corresponding histograms. Mathematically, this was evaluated as $\sum_{i=1}^{N} min(A_i^{t_a}, A_i^{t_b})$ where A is the histogram feature containing N bins, i is the bin index, and t is the indicator of the timestamp. Overlap of 1 would indicate that the histograms are identical, and vice versa. Steps followed as a part of this technique explained above are summarised in Figure 5.10.

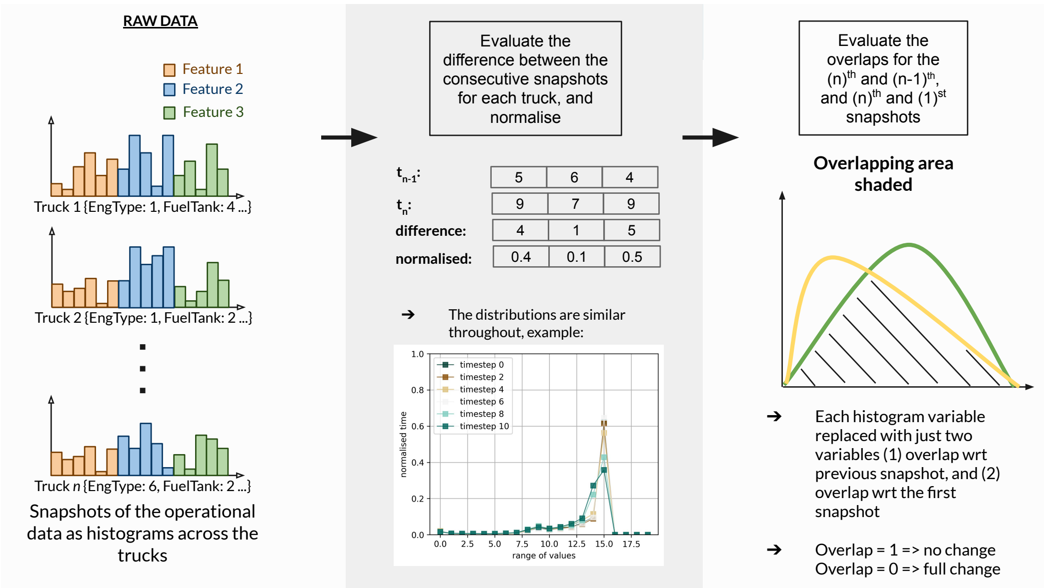
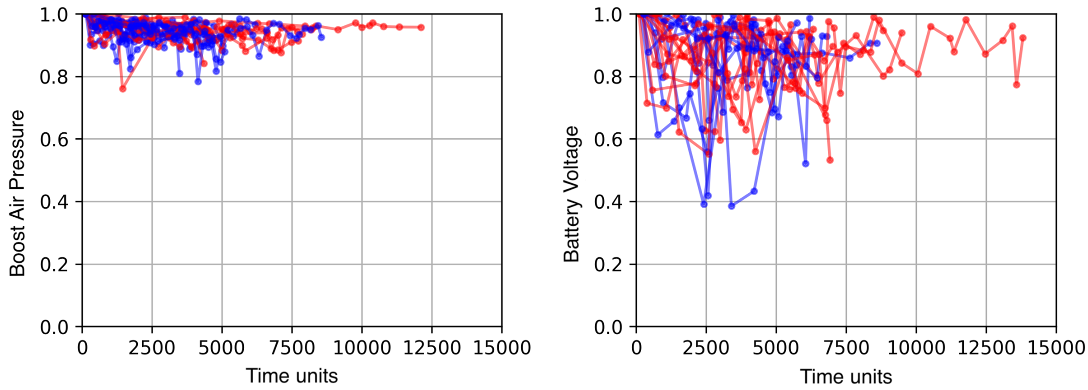


Fig. 5.10 Flowchart describing the steps followed to transform the histograms into point values by comparing the overlaps.

Figure 5.11 presents the implementation of the above steps for transforming the time-series of one of the operating parameters recorded as histograms, for both turbocharger and alternator fleets. The implementation is shown for a randomly sampled subset of 20 trucks, comprising of both healthy trucks and the trucks that encountered component failures, from each of the fleets. Time-series of operations data for the healthy trucks is shown for comparison with the failure trajectories.



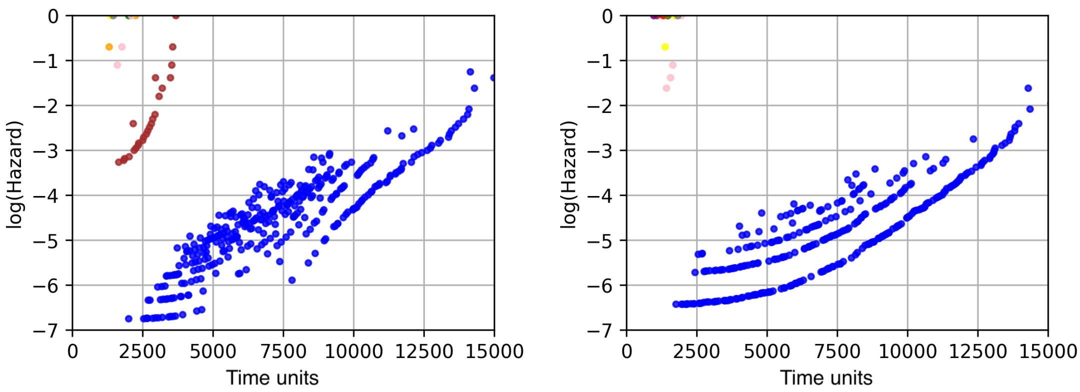
(a) Turbocharger (b) Alternator

Fig. 5.11 An example of two operating parameters transformed by evaluating the overlap with the initial histogram (for the example corresponding to turbocharger) and with the antecedent histogram (for the example corresponding to alternator)

The clustering technique described in Section 4.3 was then implemented to identify the clusters of similarly deteriorating trucks using the operations data. Given the variation in the time-series of the transformed histograms, a third degree polynomial was fit to each of the two overlaps evaluated across the operating parameters. This was followed by implementing PCA to reduce the number of features (coefficients of the polynomials), and finally the k-means clustering to identify the clusters of similarly degrading trucks.

The number of components chosen after PCA were such that they explained more than 99.5% variance in the data, and the number of clusters for k-means clustering were iteratively increased until the reduction in overall inertia was less than 10%. These stopping conditions are same as the implementation in Section 4.3. Total 11 PCA components and 7 clusters were used for the case of turbocharger failures, and 10 PCA components and 6 clusters were used for the case of alternator failures.

The empirical log-hazard curves obtained for each of the clusters, for the turbocharger and alternator fleets, using this technique are presented in Figure 5.12 where the log-hazard curves for every cluster are shown with a separate colour.



(a) Turbocharger (b) Alternator

Fig. 5.12 Empirical log-hazard curves obtained from the first clustering technique.

5.3.2 Using the Centres-of-mass of the Histograms

Although representing the trends in a failure trajectory, a drawback of the technique presented in Section 5.3.1 was that it could evaluate the similarities across the histograms only with respect to the given time-series. In other words, the failure trajectories transformed using the technique in Section 5.3.1 could not be used to compare across the trucks, as the trucks could have different initial operating regimes and therefore different representations of a standard healthy histogram. Since the histograms of the time-series are compared with the initially and the previously recorded histograms, the differences in the initial operating regimes of two trucks couldn't be incorporated.

The technique presented in this section involves reducing the differences across the consecutive histograms to a single point value, by evaluating their centres-of-mass along the x-axis. Differences between the consecutive histograms were calculated to represent the operating regimes of the trucks between the time intervals of the corresponding the histograms. Evaluating the centres-of-mass of the differences across the consecutive histograms was then equivalent to using a moving window to calculate the average value of that operations parameter, where the width of the moving window corresponds to the time difference between the consecutive histograms.

The first step, same as the technique presented in Section 5.3.1, was to calculate the differences in the bin values of the consecutive histograms comprising the time-series. The ranges of the corresponding histogram bins were normalised in the next step from $[0,1]$. The lower limit of the first bin of the histogram of a given operating parameter was kept to 0 and the upper limit of the last bin at 1, with the ranges in between corresponding to the equal bin widths. For example, if a given histogram had four bins, then the normalised bin ranges would be [0,0.25], [0.25,0.5], [0.5,0.75], and [0.75,1]. Bin widths of a histogram with normalised range of the corresponding parameter were therefore calculated as $\frac{1}{n}$, where n represents the number of bins.

The above obtained histogram differences were reduced to their corresponding point values by evaluating the centre of masses as:

$$H_{COM} = \frac{\sum_{i=1}^{n}(H_i) * (M_{H_i})}{\sum_{i=1}^{n} H_i}$$

Where H_{COM} and H_i are the centre of mass and the value of the i^{th} bin of the histogram H comprising of n bins, and M_{H_i} is the centre of the range of the i^{th} bin of the histogram.

An example of histogram and its corresponding centre of mass is shown in Figure 5.13, and transformation of a time-series of histograms into their corresponding point values is shown in Figure 5.14. An example of reducing an operating parameter to point values is shown for four trucks in the fleet in Figure 5.14, where the transformation of the histogram time-series is shown for one truck, and similar transformation for the remaining three trucks is shown without the representation of their corresponding histograms.

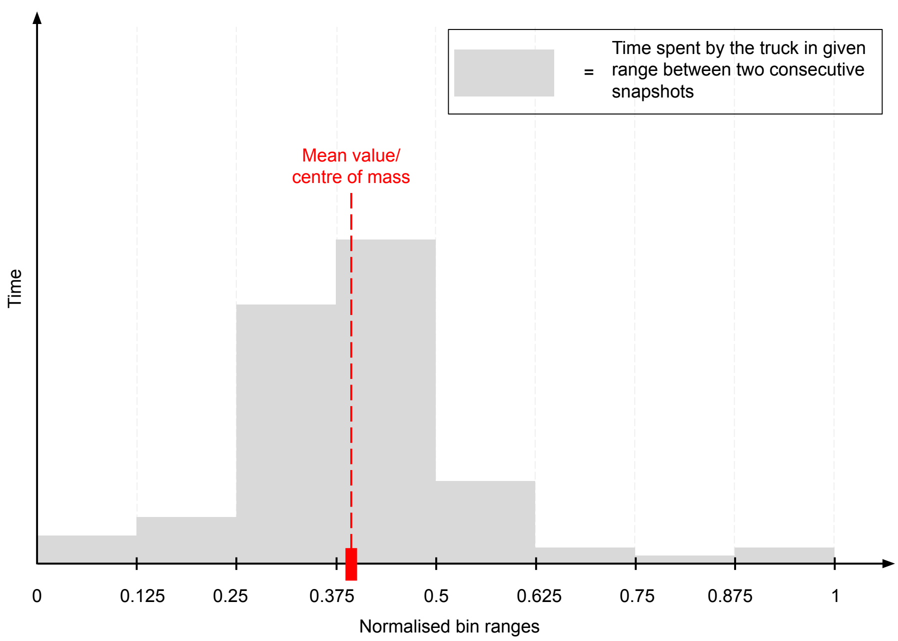


Fig. 5.13 An example of histogram and its corresponding centre of mass.

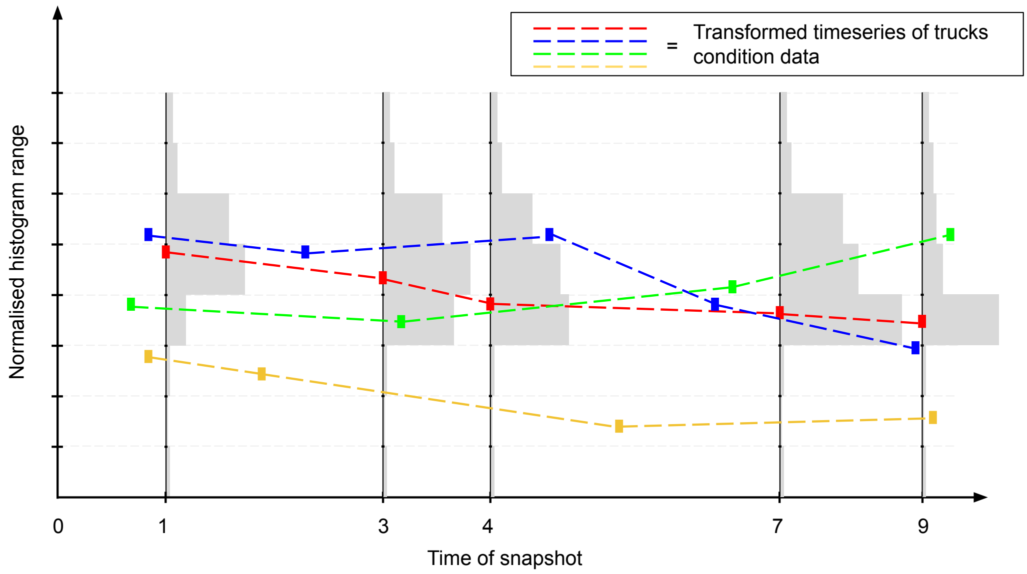


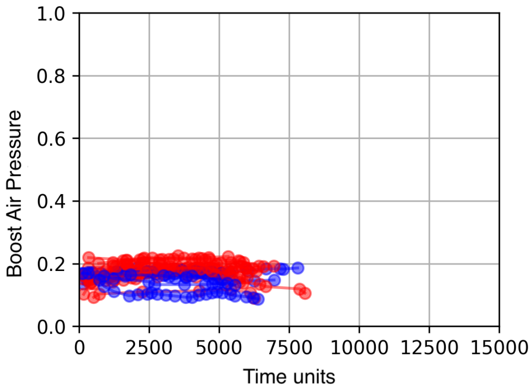
Fig. 5.14 Transformation of a time-series of histograms into their corresponding point values, shown for four trucks as an example.

Figure 5.15 presents the implementation of the above steps for transforming the time-series of one of the operating parameters recorded as histograms, for both turbocharger and alternator fleets. In this case, every recorded histogram was reduced to a single value (being its corresponding centre-of-mass). The implementation is shown for a randomly sampled subset of 20 trucks, comprising of both healthy trucks and the trucks that encountered component failures, from each of the fleets. Time-series of operations data for the healthy trucks is shown for comparison with the failure trajectories.

The clustering technique described in Section 4.3 was then implemented to identify the clusters of similarly deteriorating trucks using the operations data. Given the variation in the time-series of the transformed histograms, a third degree polynomial was fit to each of the two overlaps evaluated across the operating parameters. This was followed by implementing PCA to reduce the number of features (coefficients of the polynomials), and finally the k-means clustering to identify the clusters of similarly degrading trucks.

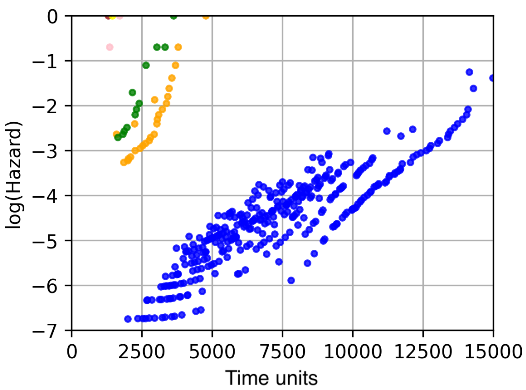
The number of components chosen after PCA were such that they explained more than 99.5% variance in the data, and the number of clusters for k-means clustering were iteratively increased until the reduction in overall inertia was less than 10%. These stopping conditions are same as the implementation in Section 4.3. Total 8 PCA components and 6 clusters were used for the case of turbocharger failures, and 8 PCA components and 6 clusters were used for the case of alternator failures.

The empirical log-hazard curves obtained for each of the clusters, for the turbocharger and alternator fleets, using this technique are presented in Figure 5.16 where the log-hazard curves for every cluster is shown with a different colour.



(a) Turbocharger (b) Alternator

Fig. 5.15 Examples of the histogram time-series transformed into point values for one operating parameter.



(a) Turbocharger (b) Alternator

Fig. 5.16 Empirical log-hazard curves obtained from the second clustering technique.

5.3.3 Using the Specifications Data

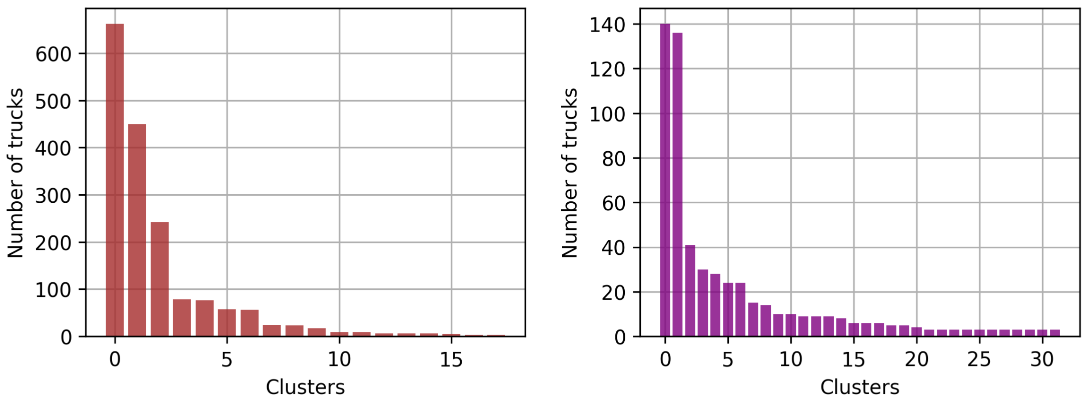
The third technique used for identifying the clusters in the Scania dataset was to use the technical specifications. The technical specifications provided in the Scania dataset reflect the customisation opted for the corresponding trucks. And because the customisation is chosen based on the anticipated usage of the trucks by their owners, clustering based on the technical specifications provides for the information needed to identify the clusters of similarly operating trucks.

The names and the cardinalities of the technical specifications used for clustering the failure trajectories for both alternator and turbocharger failures are mentioned in Table 5.2. It should be noted that the grouping is unique to this study, where the trucks are believed to be operating similarly if they share the same combinations of the technical specifications. Given the large number of combinations possible for the set of specifications, the failures observed in only those clusters were considered which had more than one truck.

Table 5.2 Technical specifications and corresponding cardinalities for clustering similarly operating trucks.

Turbocharger specification	**Cardinality**	**Alternator specification**	**Cardinality**
Cab type	3	Cab type	3
Engine type	6	Battery system type	2
Exhaust break	1	Battery	4
Exhaust outlet direction	4	Auxiliary heating	5
		Engine type	6
		Alternator charge	3

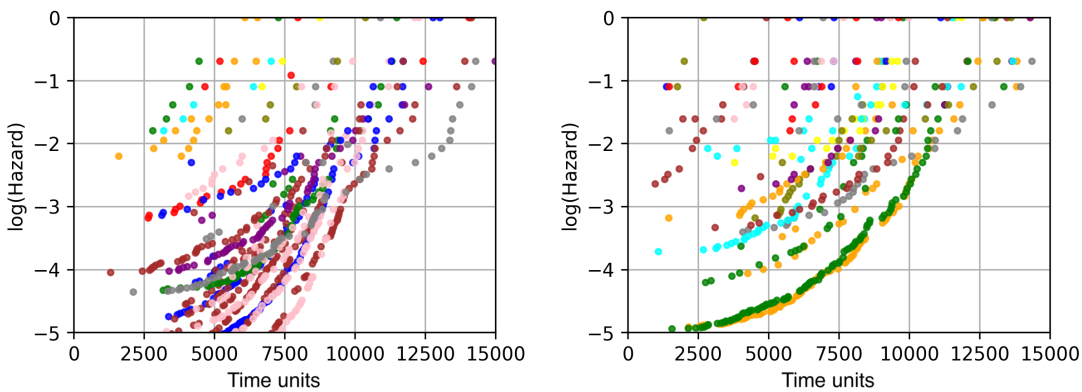
Figure 5.17 presents the number of trucks corresponding to the possible combinations of the specifications, for the failures observed in the turbocharger and alternator fleets. Cluster ids shown along the x-axis correspond to the possible combinations of the technical specifications. Figure 5.18 presents the empirical log-hazard curves obtained from the trucks clusters identified using the specifications data.



(a) Turbocharger (b) Alternator

Fig. 5.17 Number of trucks corresponding to the distinct possible combinations of technical specifications. Each point along the x-axis represents a combination, and only those combinations representing at least 2 trucks are used as clusters.

In Figure 5.17 it is observed that a relatively high number of trucks belong to only the first three and the first two combinations for turbochargers and alternator failures respectively. The plots in Figure 5.17 show that majority of the clusters do not contained sufficient number of observed failures, and therefore must rely on collaborative prognosis.



(a) Turbocharger (b) Alternator

Fig. 5.18 Empirical log-hazard curves obtained from the third clustering technique.

5.3.4 Comparing the Clustering Techniques

The clustering suitable for the Scania dataset was identified by comparing the empirical log-hazard curves presented in Figures 5.12, 5.16, and 5.18 corresponding to the three clustering techniques. Drawing from the experiment presented in Section 4.4.1, a good clustering algorithm is characterised by identifying sub-fleets with minimal diversity in the deterioration rates of the comprising clusters. This is reflected in the log-hazard plots as narrower ranges in the times-to-failures and not being split into multiple functions.

It is observed in Figures 5.12, 5.16, and 5.18 that using specifications data for clustering results in the empirical log-hazard plots with minimum ranges in the corresponding times-to-failures, and also the log-hazard plots corresponding to a single cluster are not split into multiple functions. In the log-hazard plots of Figures 5.12 and 5.16, the clusters obtained are highly skewed in the sense that most trajectories are clustered together, resulting in negligible reduction in the ranges of the time-to-failures of the comprising trucks and also the log-hazard curves being split into multiple functions. As such, the clusters obtained using the specifications data of the trucks are used for further analyses presented herewith.

5.4 Modelling the Failures in Scania Dataset

This section presents the implementation of the fleet-wide, cluster-specific, and hierarchical Weibull density models for the Scania dataset, with the formulations same as in Sections 4.5 and 4.6. The models are briefly summarised in Section 5.4.1.

5.4.1 Model Formulation Summaries

Consider a fleet comprising of $K \in \mathbb{Z}+$ asset clusters indexed as $\{1,2,...,k\}$, each comprising of $I_K \in \mathbb{Z}+$ assets indexed as $\{1,2,...,i\}$. The probability of failure $f(t_{(i,k)}, \alpha_k, \beta_k, \gamma_k,)$ at time t for the i^{th} asset in k^{th} cluster is given according to a 3-parameter Weibull distribution as:

$$f(t_{(i,k)}, \alpha_k, \beta_k, \gamma_k,) = \frac{\alpha_k}{\beta_k}\left(\frac{t-\gamma_k}{\beta}\right)^{\alpha_k - 1} e^{-\left(\frac{t-\gamma_k}{\beta_k}\right)^{\alpha_k}} \tag{5.1}$$

Where α, β, and γ are the shape, scale, and location parameters of the Weibull distribution. The location parameter (γ) is not considered here as the starting point of operations are all at $t = 0$.

The fleet-wide model involves *complete pooling* of the data, and ignores the diversity of degradation behaviours observed across the fleet. The Weibull parameters in the case of the fleet-wide model are inferred by treating the entire fleet as a single cluster and placing a non-informative uniform prior over the Weibull parameters as:

$$\begin{aligned} \alpha &\sim \mathcal{N}(0, 1000) \\ \beta &\sim \mathcal{N}(0, 1000) \\ \mathbf{T} &\sim Weibull(\alpha, \beta) \end{aligned} \tag{5.2}$$

In the case of cluster-specific models however, it is believed there exists independent Weibull distributions from which the times to failures of all the assets in that cluster are sampled. Nevertheless, non-informative priors are used for the Weibull parameters. Let $\mathbf{T}_k$ denote the array of times to failures observed in the cluster k, which are sampled from the corresponding independent Weibull distributions as:

$$\begin{aligned} \alpha_k &\sim \mathcal{N}_k(0,1000) \\ \beta_k &\sim \mathcal{N}_k(0,1000) \\ \mathbf{T_k} &\sim Weibull(\alpha_k,\beta_k) \end{aligned} \tag{5.3}$$

The hierarchical model is an extension of the independent cluster-specific models, where two Normal distributions are defined at a higher level from which the Weibull parameters of the clusters are sampled. This is in contrast to using non-informative priors and enables learning across the clusters. Concretely, the α_k and β_k parameters of the clusters are each sampled from their corresponding Normal distributions at a higher level which are common to all the clusters present in the fleet. The hierarchical Weibull model is shown as a block diagram in Figure 5.19 and mathematically described as:

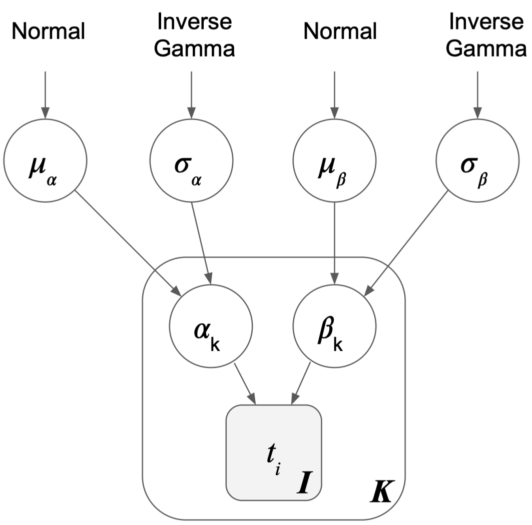


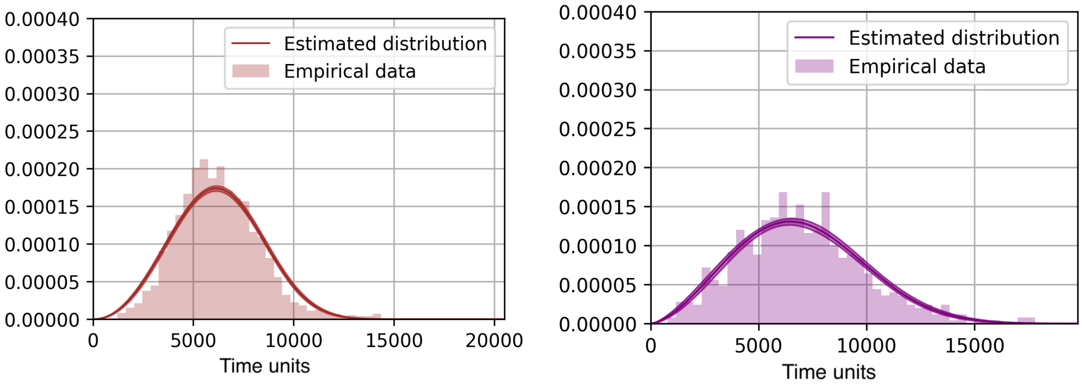
Fig. 5.19 Block diagram of the hierarchical Weibull model.

$$
\begin{aligned}
&\mu_\alpha \sim \mathcal{N}(0, 1000)\\
&\sigma_\alpha \sim InverseGamma(1, 1)\\
\\
&\mu_\beta \sim \mathcal{N}(0, 1000)\\
&\sigma_\beta \sim InverseGamma(1, 1)\\
\\
&\alpha_k \sim \mathcal{N}_k(\mu_\alpha, \sigma_\alpha)\\
&\beta_k \sim \mathcal{N}_k(\mu_\beta, \sigma_\beta)\\
\\
&\mathbf{T_k} \sim Weibull(\alpha_k, \beta_k)
\end{aligned}
\tag{5.4}
$$

For implementing the fleet-wide model, the times-to-failures of all the trucks were treated as one single cluster thus pooling all the observed times-to-failures. However, it was observed in the empirical log-hazard curves corresponding to the clustering techniques, shown in Section 5.3, that clustering the fleet using technical specifications was deemed most suitable. The corresponding clusters obtained for turbocharger and alternator failure trajectories were used for implementing the cluster-specific and hierarchical models for the results presented in this section.

5.4.2 Model Implementations

Figure 5.20 shows the results obtained when the fleet-wide model was implemented for modelling the observed times-to-failures for the cases of turbocharger and alternator failures. The background plot in Figure 5.20 represents the distribution of the times-to-failures across the fleet as a histogram, and on top of which is plotted the mode and the central 80-percentile of the inferred Weibull density distributions.



(a) Turbocharger (b) Alternator

Fig. 5.20 Fleet-wide Weibull model inferred for the times-to-failures observed in turbocharger and alternator fleets.

The results obtained from implementing the cluster-specific vs hierarchical Weibull models are shown in a similar manner in Figures 5.21 and 5.22 for the turbocharger and the alternator failures respectively. The cluster-specific vs hierarchical model inferences shown together on the corresponding plots for the ease of comparison.

Given the large number of clusters in fleets corresponding to either targeted components, only 12 clusters with fewest number of observed failures are shown. Nevertheless, all the clusters were used in the inference process and the performances of the cluster-specific and the hierarchical models were seen to converge as the number of observed failures increased (ref. Section 4.7 for a comprehensive experiment on the impact of number of observed failures conducted with the simulated dataset).

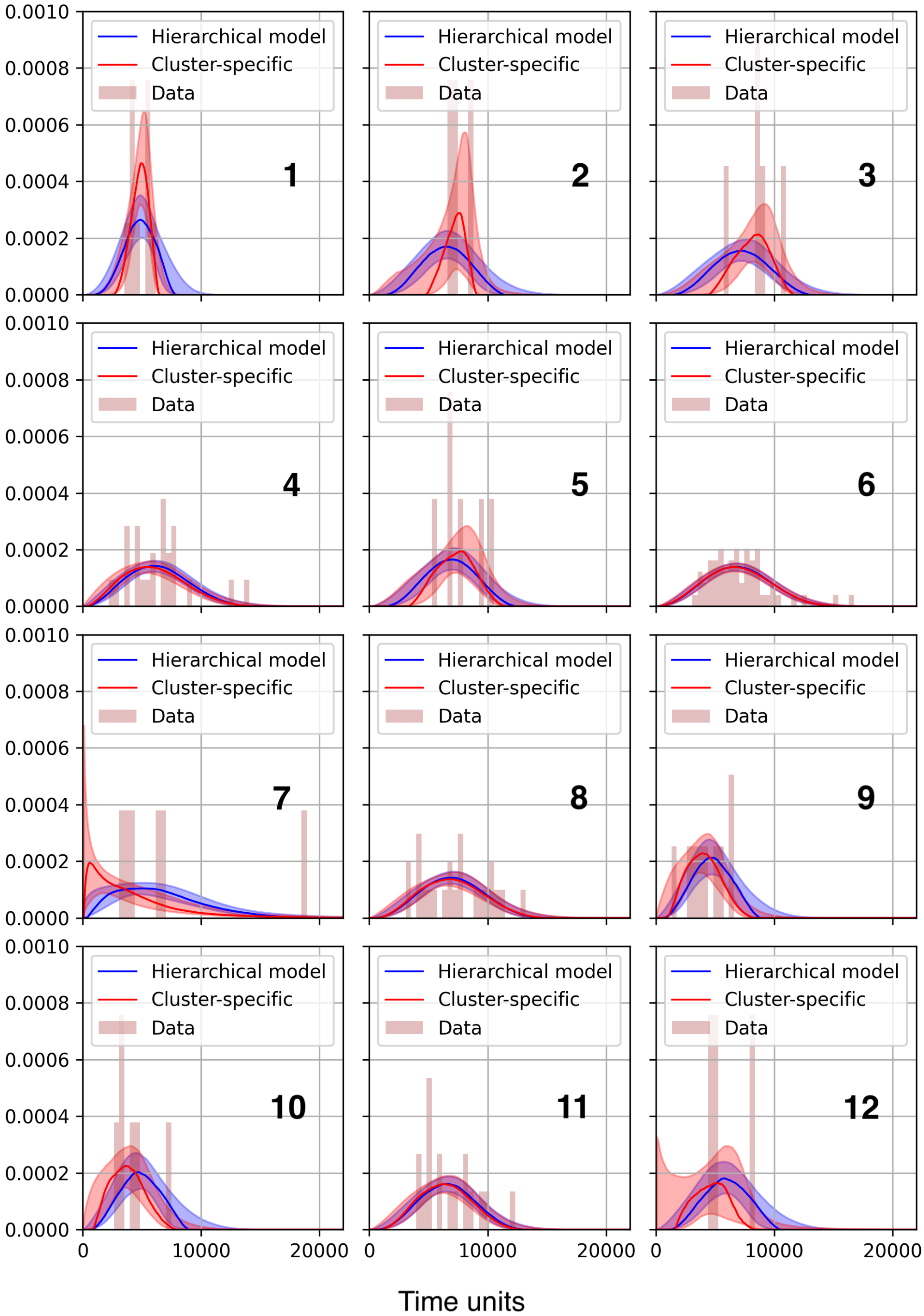


Fig. 5.21 Cluster-specific and hierarchical Weibull models inferred for the clusters in the turbocharger fleet.

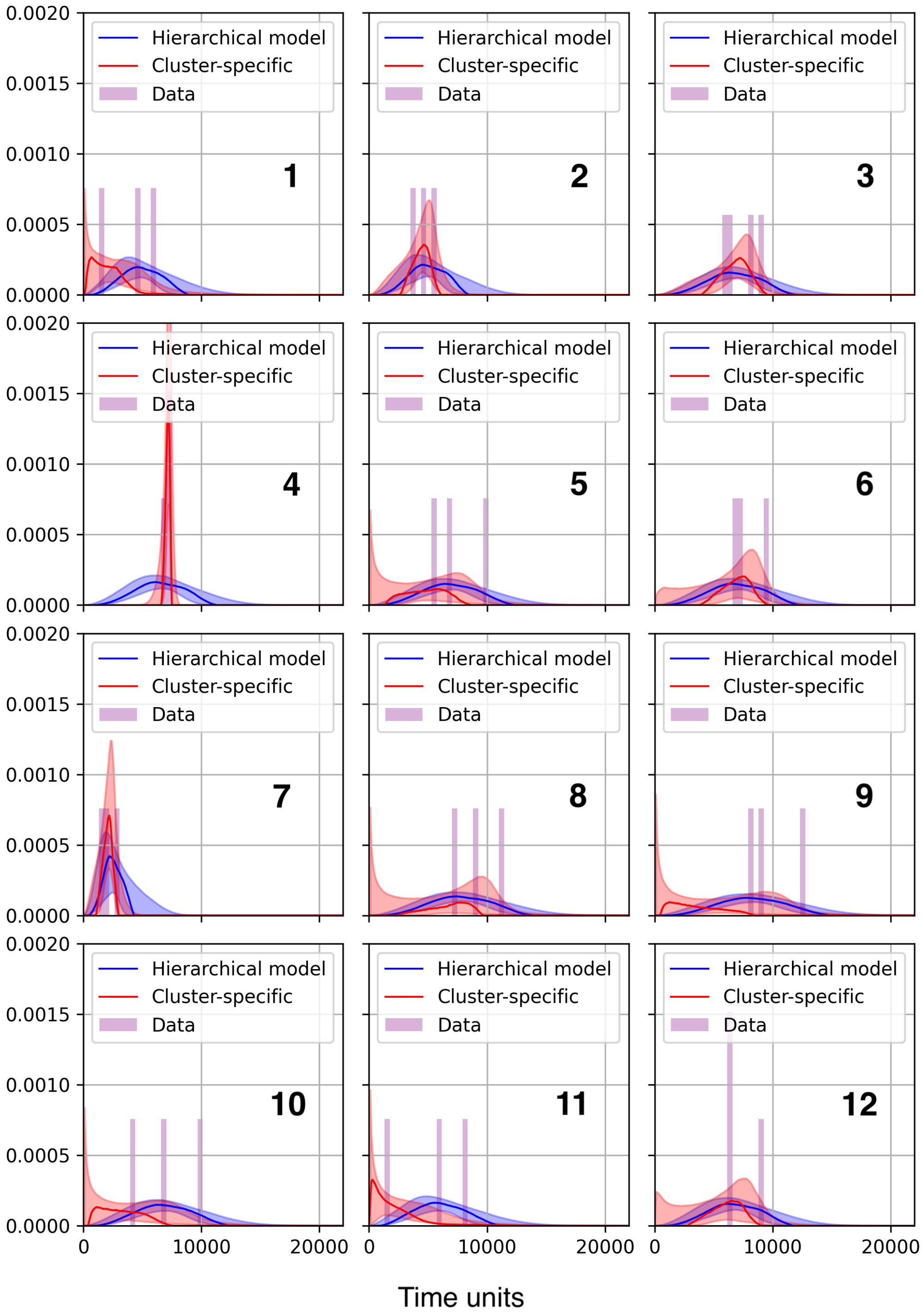


Fig. 5.22 Cluster-specific and hierarchical Weibull models inferred for the clusters in the alternator fleet.

The fleet-wide, cluster-specific, and hierarchical Weibull density models were implemented for the Scania dataset in this section. The Weibull parameters in all three models were associated with the same family of non-informative priors. The priors however were modelled jointly for the hierarchical model and independently for the cluster-specific models. It can be observed for the case of fleet-wide model in Figure 5.20 that the parameters are associated with low uncertainty as the central 90-percentile inference shown by the shaded region is extremely close to the mode. This is due to large amount of data pooled from all the trucks. However, the times-to-failures across the fleet vary over a wide range causing the inferred distribution itself to be highly spread out. On the other hand, the cluster-specific distributions presented in Figures 5.21 and 5.22 tend to be relatively specific given a smaller range of times-to-failures for a given cluster compared to the entire fleet. But the drawback associated with the cluster-specific models is that the inferred distributions are associated with a high uncertainty, due to presence of fewer number of observed failures. The inferences are also biased when the time-to-failures observed in the corresponding cluster, leading to the distributions with exceptionally low variance.

The above observation is true specifically for the clusters **2**, **3**, **5**, **7**, and **12** in Figure 5.21, and nearly all the clusters in Figure 5.22 where using hierarchical model is recommended due to fewer number of observed failures. The distributions inferred from the hierarchical model are more certain compared to those inferred using the cluster-specific models as they incorporate the knowledge from the rest of the fleet as well as are able to account for the smaller range of times to failures observed in a cluster. For example the distributions inferred using a hierarchical model for cluster **7** vs that inferred for cluster **5** in Figure 5.22. For the clusters with high number of observed failures, the hierarchical and cluster-specific models tend to converge like it is observed in clusters **8** and **11** in the turbocharger fleet.

5.5 Conclusions

This section summarises the insights and challenges while implementing a hierarchical model for an industrial fleet, and for the case of Scania dataset specifically. Future research directions for prognosis of the trucks maintained by Scania is also provided in the following points:

1. A high prevalence of clusters are observed within the Scania fleet, most of which are associated with insufficient number of failures for independent cluster-specific models. This is often the case for most industrial fleets what comprise of diverse assets, or assets operating in diverse environments. Hierarchical model is necessary in such

instances for modelling the failures as the fleet-wide model is associated with high bias whereas the cluster-specific models are associated with a high variance.

2. Clustering is critical for achieving optimal performance with the hierarchical model. Asset condition data provides information about similarities across the assets within a fleet, but other sources of information such as expert knowledge or technical specifications should not be ignored, like in the current case study. For the Scania dataset the asset clusters could not be identified using the operating parameters and the techniques presented herewith, but the technical specifications provided for a straightforward way to reasonably cluster the assets. It is concluded from the estimated Weibull density functions describing the times-to-failures presented in Section 5.4.2 that the hierarchical model is able to reduce the variance associated with the independent models for an industrial dataset.

3. The empirical log-hazard curves presented in Section 5.3 show that meaningful clusters could not be obtained using the operations data corresponding to the first two techniques, and therefore the clusters obtained using the technical specifications were deemed suitable for modelling the observed failures corresponding to the third technique. The main challenges while using the operating parameters from the Scania dataset for clustering the failure trajectories was the histogram nature of the data and irregularities across the time-series. The transformed time-series presented in Figures 5.15 show that the values of the histograms across the trucks and the failures were concentrated around the same range throughout the trajectories with minimal trends. Similarly in Figure 5.11 as well no significant trends were noticed in the intersection across the consecutive histograms and with the initial healthy-regime histogram.

4. In the case study presented herewith, the condition data recorded from the trucks were stored as histograms. Condition data recorded from the assets often exists in various formats including continuous point values, categorical warning levels, discrete events, or even human notes. While storing the asset condition data, the industries must be aware of the information vs storage costs trade off. For the current case specifically the storage costs decline as the number of bins and the frequency of the snapshots are reduced, but at the same time it would not show any trends in the failure trajectories.

5. An important future research direction for prognosis of the trucks maintained by Scania is to explore techniques for clustering the failure trajectories using operating parameters, as the clusters obtained using technical specifications do not provide real-time condition and RUL estimations for the trucks. When a new truck is introduced, its

technical specifications can be used only to get a static estimate of the RUL based on which cluster the truck belongs to. However, it is shown in Chapter 4 that a hierarchical model can be implemented for real-time prognosis of an asset using the condition data, as the clusters in the fleet are continuously updated as more data is acquired. Similar implementation is possible for Scania is the operating parameters representing the real-time conditions of the trucks are used for clustering.

Chapter 6

Conclusions and Future Research Directions

This chapter presents the general conclusions to this thesis along with a summary of the academic contributions, limitations of the proposed techniques, and the future research directions. The following chapter is structured as: Section 6.1 presents a recap of the research questions introduced in Chapter 1, and discusses the results presented in this thesis in the context of the research questions. Section 6.2 presents a summary of the academic contributions in the form of the peer-reviewed articles published or under review during the course of this research. The shortcomings of the techniques presented in this thesis are listed in Section 6.3, and the future research directions are presented in Section 6.4.

6.1 General Conclusions

This section presents a recap of the research questions introduced in Chapter 1, and discusses the results presented in this thesis in the context of the research questions.

Chapter 1 highlighted that collaborative prognosis is critical for data-driven prognosis because industrial fleets are non-ergodic systems comprising of assets operating in diverse conditions. While modelling the condition data for prognosis applications, industries either rely on a single fleet-wide model trained using all the data pooled together or on independent models trained using the isolated data from a single asset. Both these techniques are unsuitable for prognosis applications because the fleet-wide model is bound to be associated with a high bias, whereas the independent clusters/ assets-specific models are associated with high variance especially if an asset has sparse data [104, 155]. It was also explained in Chapter 1 that anomaly detection is an essential element of the data-driven prognosis pipeline, and a similar challenge exists for the prevalent anomaly detection techniques in the industries.

The problem introduced in Chapter 1 was further supported and analysed through a literature review presented in Chapter 2. It was explained in Chapter 2 that if an asset has insufficient failure locally available data, it is beneficial for that asset to rely on other similar assets in the fleet for health estimation. Collaborative prognosis is a technique that mitigates the problem of sparse data by identifying clusters of similarly operating assets and learning within the asset clusters. The advancements in the internet, computation, and communication technologies have enabled distributed control and implementation of collaborative prognosis in real time. However, a critical research gap exists due to lack of a systematic technique for collaborative prognosis that enables knowledge transfer across the asset clusters. It is also outlined in this chapter that statistical hierarchical modelling is a systematic technique for enabling collaborative prognosis. Applications of statistical hierarchical modelling across the industries for addressing similar problems are also discussed.

To that end, this thesis contributes towards bridging the research gap for a technique to enable collaborative prognosis in industrial fleets. The following points summarise the research questions outlined in Chapter 1 and the chapters contributing towards the corresponding research questions:

1. **Research Question 1:** How to model the asset fleet data systematically to enable collaborative prognosis and anomaly detection? Statistical hierarchical modelling is identified and proposed in this thesis (in Chapter 2) as a solution for enabling collaborative prognosis. The use of the hierarchical model is demonstrated for collaborative anomaly detection (in Chapter 3), collaborative prognosis (in Chapter 4), and modelling the times-to-failures in an industrial dataset (in Chapter 5).

2. **Research Question 2:** How effective is the technique of statistical hierarchical modelling for collaborative prognosis and anomaly detection? This research question aims at exploring the added advantages or drawbacks of using a hierarchical model for collaborative prognosis, both for simulated and a real industrial dataset. The proposed technique of hierarchical modelling is analysed in this thesis for real-time collaborative anomaly detection and collaborative prognosis (in Chapter 3) applications in simulated datasets. Moreover, an industrial dataset was also used to study a case for modelling the observed times-to-failures using a hierarchical model (in Chapter 5).

Statistical hierarchical models are characterised by multi-level modelling. The data are sampled from the lower level models and parameters of the lower level models are in turn sampled from the shared higher level models [53]. For the collaborative prognosis applications, the lower level models model the data corresponding to each asset or a sub-fleet of similarly operating assets. A hierarchical model encourages learning for those assets with sparse data as the higher level models incorporate prior information about the general behaviour observed in other similar assets comprising the fleet [30].

In Chapter 3 a hierarchical model of multi-variate Gaussian is proposed and demonstrated for anomaly detection in a simulated fleet of assets. The asset data were simulated for the experiments from the underlying Gaussians characterised by clusters with different sets of means and variances in the multi-dimensional space. This represented a non-ergodic fleet of assets, and the actual cluster of the assets was unknown to the hierarchical model. A latent variable was introduced that denoted the predicted cluster of the assets, and the model parameters were optimised using the expectation maximisation algorithm. Bhattacharyya distance was used to compare the true parameter values across the assets with those estimated using (1) hierarchical model, (2) asset-specific models, and (3) a fleet-wide model. The experimental results, in Figure 3.9, showed that the hierarchical model was significantly better in the early periods of the asset operations when sufficient data were not available locally, achieving more than 50% reduction in the Bhattacharyya distance compared to the independent models. While both independent and hierarchical model estimates converged to the true parameter values, the fleet-wide model failed to converge and showed highest variance in the classification accuracies evaluated across the fleet for the testing dataset, shown in Figure 3.6. It was also concluded from the experimental results shown in Figure 3.6 that the advantage of the hierarchical model depends on the proportion if the low-data assets in the fleet.

Chapter 4 presented a hierarchical Weibull model, where the parameters of the Weibull models used to model the times-to-failures observed in clusters of similarly operating assets shared common non-informative Gaussian priors. The hierarchical Weibull model was used to model the times-to-failures in a fleet of 200 simulated turbofans comprising the C-MAPSS dataset. The turbofans operating in the same environment and could fail in either due to high pressure compressor failure or fan degradation. The empirical log-hazard curves, shown in Figure 4.4, suggested that the turbofan fleet comprised of sub-fleets of similarly operating assets. The same was also observed when a single Weibull distribution was used to model the times-to-failures across the fleet, in Figure 4.11. The procedure for identifying similarly operating assets using the corresponding failure trajectories was described in Section 4.3. It was observed that some of these clusters were associated with sparse number of failures and therefore modelling them using independent Weibull models led to high variance, shown in Figures 4.13 and 4.14. It was observed that the variance associated with the independent Weibull models was significantly reduced especially when a cluster had sparse data, when the hierarchical Weibull model was used. This is shown in Figure 4.18 corresponding to the experiment where the number of failures observed within a cluster of assets were sequentially increased to compare the performances of the hierarchical and independent Weibull models. Another experiment was also conducted where the higher level parameters were controlled to show their effect on the lower level estimates, the results from which are presented in Figure 4.19 and 4.20. It was concluded from this experiment that the higher level parameters can be adjusted by the operators to incorporate the expert knowledge or other sources of information about the similarity across the clusters while estimating the hierarchical model parameters.

The procedure for real-time collaborative prognosis using the aforementioned hierarchical model was also presented in Chapter 4 using a simulated turbofan fleet. The procedure for implementing the hierarchical model for real-time collaborative prognosis can be summarised in the following steps: (1) preprocess, (2) cluster, (3) model, and (4) predict which are shown in Figure 4.21. Moreover, an experiment was conducted where the quality of clustering was varied to analyse its effect on the overall predictive performance of the algorithm. It was concluded that the performance of the hierarchical model critically depends on clustering. In Figure 4.25 it is observed that the errors in predictions, in terms of accuracy and the variance across the testing dataset, significantly reduce as the turbofans approach the failures. The errors in predictions in the early time-steps of operations can be mitigated if the clustering is supported by other sources of information such as expert knowledge or specifications data. The importance of clustering for collaborative prognosis using hierarchical Weibull model is further supported by the fact that the errors in the final segment in the case of threshold 0.1 (good clustering) are one-fourth of those in corresponding segment of threshold 0.5 (poor clustering) in Figure 4.26.

In Chapter 5, a case study was presented that involved modelling the times-to-failures in a fleet of long-haulage trucks. It was noted via the empirical log-hazard curves in Figure 5.12 that the fleet comprised of clusters of similarly operating trucks resulting from customisation and diversity in their operating conditions. Section 5.3 discusses three techniques used for identifying the clusters, two of which involved using the condition data for clustering. It was concluded from Figures 5.12 and 5.16 that the clusters of similarly operating trucks could not be identified using the condition data in its current form and preprocessing. The technical specifications were therefore used to identify the clusters, the empirical log-hazard curves obtained thereof are presented in Figure 5.18. Some of the identified clusters were associated with sparse number of historical failures, and therefore the hierarchical Weibull model was deemed appropriate for modelling the times-to-failures across the fleet. It was concluded in this chapter that modelling the times-to-failures using a hierarchical Weibull model reduced the variance for the clusters with sparse data, compared to modelling their times-to-failures using independent cluster-specific models for a real industrial fleet. The most challenging aspect of implementing the hierarchical model for the industrial fleet in this case study was identifying the clusters of similarly operating assets. Multiple sources of information should be used for the purpose of clustering but the operators must be aware that collaborative prognosis critically depends on identifying reasonable clusters of the assets.

6.2 Academic Contributions

This section presents a summary of the academic contributions in the course of this research. The research questions, objectives, and publications stemming from the corresponding contributions are also listed alongside:

1. **Statistical hierarchical modelling for industrial asset fleets** An extensive analysis of industrial collaborative prognosis is conducted to identify the optimal use cases and prevalent challenges. Furthermore, a survey of various learning techniques for data arising from a population shows that the technique of hierarchical modelling is a suitable solution to enable collaborative prognosis. By sharing information between similar assets, hierarchical Bayes with mixed effects improves the predictive performance of the asset-specific models. This is presented in Chapter 2, and addresses the research objective 1. This contribution has also led to the journal article [20] *Knowledge Transfer in Engineering Fleets: Hierarchical Bayes for Multi-Task Learning with Mixed Effects* in *Computer Aided and Civil Infrastructure Engineering* by *Lawrence A. Bull, Maharshi Dhada, Olof Steinert, Tony Lindgren, Ajith Kumar Parlikad, Andrew Duncan, Mark Girolami* The paper proposes a population-level analysis to addresses issues of data sparsity when building predictive models of engineering infrastructure. It is shown that parameter estimation is improved when sub-fleets of assets are allowed to share correlated information at different levels in the hierarchy. In turn, groups with incomplete data (automatically) borrow statistical strength those that are data-rich. The correlations can be inspected to inform which assets share information for which *effect* (i.e. parameter).

2. **Statistical Hierarchical Model for Anomaly Detection** A hierarchical model for anomaly detection is presented that systematically identifies similar assets and enables collaborative learning within the clusters of similar assets. It addresses the problem of anomaly detection with sparse data in asset fleets. Results obtained with the hierarchical model show a marked improvement in anomaly detection for assets having

low amount of data, compared to independent modelling or having a model common to the entire fleet. Analytical solution to estimate the model parameters using the asset data is also derived as an academic contribution. This is presented in chapter 3, addresses the research questions 1 and 2, and research objectives 2 and 3. This work also led to the journal article [30] *Anomaly detection in a fleet of industrial assets with hierarchical statistical modeling* in *Journal of Intelligent Manufacturing (2019)* by *Maharshi Harshadbhai Dhada, Mark Girolami, Ajith Kumar Parlikad*

3. **Hierarchical Weibull model for modelling times-to-failures in an asset fleet** A hierarchical Weibull model is presented for modelling the observed times-to-failures in an asset fleet. Furthermore, it is shown using a simulated fleet of assets that a hierarchical model inherently encourages collaborative learning for the clusters with sparse data. This is because it defines common prior distributions for the cluster-specific Weibull parameters at a higher level. The hierarchical model is able to mitigate the problem of high variance in cluster-specific independent models by learning from the failures occurring in the other clusters comprising the fleet. This is shown in Chapter 4, addresses the research questions 1 and 2, and research objectives 4 and 5. This work has also led to the publication of *Modelling Failures in an Asset Fleet using a Statistical Hierarchical Model* in *European Workshop on Structural Health Monitoring (2022)* by *Maharshi Dhada, Lawrence A. Bull, Mark girolami, Ajith Kumar Parlikad*

4. **Using the hierarchical Weibull model for real-time collaborative prognosis** The procedure for using the hierarchical Weibull model for real-time collaborative prognosis is presented. The procedure involves clustering the existing failure trajectories, followed by identifying the closest cluster to the operating asset. Experimental results show the effect of the clustering threshold parameter on the predictive performance and also the increasing certainty as more data is accumulated. This can be found in Chapter 4, addressing the research questions 1 and 2, and the research objectives 4 and 5. This work has also led to the submission of *Real-time Collaborative Prognosis using a Hierarchical Weibull Model* in *Reliability Engineering Safety Systems* by *Maharshi Dhada, Lawrence A. Bull, Mark Girolami, Ajith Kumar Parlikad*

6.2.1 Additional, relevant contributions during the PhD Study

Additional contributions, in the form of academic publications, published during the course of the PhD study are listed below:

1. [34] *Empirical Convergence Analysis of Federated Averaging for Failure Prognosis* in *IFAC-PapersOnLine (2020)* by *Maharshi Dhada, Amit Kumar Jain, Ajith Kumar Parlikad* This paper empirically analyses the convergence of the Federated Averaging (FedAvg) algorithm for a fleet of simulated turbofan engines. Federated Learning is an emerging technique that has recently also been proposed as a fitting solution for prognosis of industrial assets. However, even the most commonly used Federated Learning algorithms lack theoretical convergence guarantees, and therefore their convergence must be analysed empirically. Results demonstrate that while FedAvg is applicable for prognosis, it cannot acknowledge the differences in asset failure mechanisms. As a result, the prognosis framework needs to be modified such that similar failures are clustered together before FedAvg can be implemented.

2. [31] *Predicting Bridge Elements Deterioration, using Collaborative Gaussian Process Regression* in *IFAC-PapersOnLine (2020)* by *Maharshi Harshadbhai Dhada, Georgios M. Hadjidemetriou, Ajith Kumar Parlikad* This paper presents a Gaussian Process Regression (GPR) based collaborative model for predicting the condition of bridge elements with limited available inspection data per bridge. This model has been applied in 137 bridge decks, showing that collaborative prognosis has the potential to predict the condition of different types of bridge elements, composing different types of bridges.

3. [33] *Secure and communications-efficient collaborative prognosis* in *IET Collaborative Intelligent Manufacturing (2020)* by *Maharshi Dhada, Amit Kumar Jain, Manuel Herrera, Marco Perez Hernandez, Ajith Kumar Parlikad* This paper analyses the ability of Federated Averaging for collaborative prognosis and ensuring sensitive operational data is not shared between organisational boundaries. An example implementation is demonstrated for the prognosis of a simulated turbofan fleet, where federated averaging algorithm is used as an alternative for the data exchange step. The results confirm that federated averaging retains the performance of conventional collaborative prognosis while eliminating the exchange of failure data within assets. This removes a critical hindrance in industrial adoption of collaborative prognosis, thus enhancing the potential of predictive maintenance. However, the parameters need to be carefully tuned for optimal performance, and Federated Averaging is not capable of incorporating multiple failure modes in the fleet.

4. [32] *Comparison of Agent Deployment Strategies for Collaborative Prognosis* in *2021 IEEE International Conference on Prognostics and Health Management (ICPHM)* by *Maharshi Dhada, Marco Perez Hernandez, Adrià Salvador Palau, Ajith Kumar*

Parlikad This paper analyses the effects of Digital Twin deployment strategies on the effectiveness of predictive maintenance activities relying on distributed collaborative prognosis. The results show that no single architecture or deployment strategy can be deemed best across all failure rates and noise levels. The conclusion derived in this paper provides guidance to the asset owners to choose the most suitable combination for a given application.

5. **In press** *Weibull Recurrent Neural Networks for Failure Prognosis Using Histogram Data* in *Neural Computing and Applications* by *Maharshi Dhada, Olof Steinert, Tony Lindgren, Ajith Kumar Parlikad* This paper presents the first industrial use case of Weibull Time To Event Recurrent Neural Networks (WTTE-RNN) for prognosis and also a technique to preprocess the histogram data. WTTE-RNN combines the survival analyses techniques with recurrent neural networks. It is concluded in this paper that clustering is only beneficial as long as the training datasets per cluster are large enough for the correponding models to not overfit. Moreover, the censored data from assets that did not fail, are also shown to be incorporated while optimising the Weibull loss function and improve prediction performance.

6. **In press** *Population-Level Modelling for Truck Fleet Survival Analysis* in *European Workshop on Structural Health Monitoring (2022)* by *Lawrence A. Bull, Maharshi Dhada, Olof Steinert, Tony Lindgren, Ajith Kumar Parlikad, Mark Girolami* This paper stems from the work presented in chapter 5, addressing the research question 2, and the research objective 6. Population-level modelling is used in this paper to address issues of data sparsity in the survival analysis of a truck fleet. Specifically, hierarchical Bayes with mixed effects improves the predictive capability of hazard models. A set of correlated functions is learnt over the vehicle population, in a combined model, to approximate fleet predictors. Model uncertainty is reduced when sub-fleets of vehicles are allowed to share correlated information. In turn, vehicle groups with incomplete data (automatically) borrow statistical strength from data-rich groups.

6.3 Limitations of the Proposed Techniques

This section presents the limitations of the techniques presented in this thesis, which are listed below:

1. The hierarchical multi-variate Gaussian model for anomaly detection presented in Chapter 3 assumes that the asset operations are static, in the sense that the means/ variances of the asset operations do not change during the operations. The proposed

model is therefore not applicable for the scenarios where the assets perform cyclic or multiple operations.

2. The hierarchical multi-variate Gaussian model requires the number of underlying asset clusters in the fleet to be known a priori. The operators are often able to determine this based on the expert knowledge or the number of clusters can be optimised similar to the procedure presented in Section 4.3. Nevertheless, the hierarchical model is not a standalone solution for a fleet of assets for which no information is available about the asset operations.

3. The hierarchical Weibull model introduced in Chapter 4 considers the clustering and preprocessing steps separately, unlike the hierarchical multi-variate Gaussian model where the clusters and the hierarchical model parameters are jointly optimised. As such, the Weibull model does not provide an end-to-end solution for collaborative prognosis. The performance of the model critically relies on the clustering, and the operators must ensure that the asset clusters are reasonably determined for the given application. An example of this can be seen in Chapter 5 where the clusters of assets could not be satisfactorily determined using the condition data, and therefore the technical specifications had to be used for determining the asset clusters.

4. For both hierarchical multi-variate Gaussian and hierarchical Weibull models, implementing them in real-time involves re-evaluating all the steps from clustering until prediction. In other words, when a new data point is obtained for the operating assets, the hierarchical model parameters are not updated but rather re-evaluated. This is computationally expensive and therefore a critical drawback of the proposed technique in the long run.

5. Lastly, it should be noted that the proposed methodologies of using the hierarchical statistical models for industrial collaborative prognosis go hand-in-hand with expert knowledge. The underlying idea is to identify similar assets or asset clusters, and enable information sharing amongst them. Hierarchical modelling of the data enables information sharing, however *similarity* across the assets in most cases is highly subjective depending on the application. It can be noted that in Chapter 3 Figure 3.4 that relying solely on the data can result in sub-par clustering, and expert knowledge is required to assist the same. Similarly also for the application in Chapters 4 where the subjective nature of clustering and its effect on the prediction performance is observed in Figures 4.5b and 4.26 respectively. Lastly, the industrial case study presented in Chapter 5 also discusses in Section 5.3 that often for the industries no

statistical method can identify the clusters and the expert knowledge must be used for clustering. Clustering using the expert knowledge can rely on for example the make or types of assets, or on their operating conditions. This knowledge can be incorporated in the higher level distributions of the hierarchical model, such that a higher variance corresponds to a fleet comprising of assets with minimal similarity and vice versa. Figures 4.19 and 4.20 explain this concept where the effect of the higher level distribution is shown for increasing variance.

6.4 Future Research Directions

This thesis proposes that statistical hierarchical modelling is a systematic technique to realise collaborative prognosis for the industrial fleets. The research presented herewith paves way for exciting extensions in various directions, which are listed in the following points:

1. An immediate extension of the research presented in this thesis is to develop dynamic versions of the hierarchical models presented herewith. A dynamic hierarchical model shall involve *updating* the model parameters rather than *re-evaluating* them as in the current implementations. This should significantly reduce the computational complexity and in turn the costs for managing large asset fleets.

2. A realistic extension of the hierarchical multi-variate Gaussian model for anomaly detection presented in Chapter 3 would be to incorporate the fact that the asset operations are not static. This can be done by using a Kalman filter in conjunction with the multi-variate Gaussians. Kalman filters excel at learning the cyclic and serial patters, which are often the case for industrial asset operations [26].

3. The example implementation in Chapter 3 was shown using a simulation fleet of assets. An interesting follow up work would be to analyse how the hierarchical anomaly detection model performs for a real world fleet of assets. Such analysis can also analyse the extent of improvement in overall maintenance cost to the organisation. Moreover, the real world implementation would enable including the categorical data for clustering the assets and improve the accuracy of the EM algorithm. This is explained in Section 3.2.2.

4. An important conclusion from the experiments presented in Chapter 3 was that a low data category asset benefits the most from the hierarchical model. Moreover, that asset has nothing to contribute towards the general fleet knowledge. Therefore, it would be interesting to analyse how a hierarchical model would perform if only the medium and

high data category assets were allowed to contribute to the higher level distributions, whereas the low data category assets only learn from them.

5. The clustering step of the hierarchical Weibull model introduced in Chapter 4 can be implemented using techniques such as Dirichlet-processes Gaussian mixture models to evaluate the number of sub-fleets of similarly operating assets [128]. Clustering and prediction steps can also be jointly implemented, like in [124] for neural network models. However, it should be noted that different formats of asset condition data require different clustering algorithms to identify the sub-fleets of similarly deteriorating assets.

6. Future research should focus on implementing the proposed hierarchical models in a distributed setting for a fleet of assets using the multi-agent system architectures. Using distributed multi-agent system architectures enhances the Distribution, Flexibility, Adaptability, Scalability, Leanness, and Resilience of collaborative prognosis for the industrial fleets [136].

7. While this thesis focuses on evaluating the accuracy and confidence of the predictions using hierarchical models, the future research can extend this to evaluating the overall business cost of the predictive maintenance pipeline relying on the hierarchical models. This involves evaluating the cost of operations incorporating the anomaly detection, prognosis, maintenance planning, and maintenance resources.

8. Future research should also focus on evaluating the value of information of obtaining and storing the condition data from the assets. As shown in Chapter 5, the industries often optimise the data storage vs the information costs, which could result in sub-optimal performances of the prognosis algorithms. An interesting future research direction is therefore to evaluate the impact of reducing the granularity of the asset condition data on the prognosis ability.

9. It should be noted that the procedure for clustering and real-time collaborative prognosis presented in Chapter 4 assumes maintenance-free operations of the industrial assets. This assumption is made to resemble a stable failure trajectory of the assets, so that the asset deterioration depends only on the systemic wear and tear, and can be modelled as a single function. In case of a maintenance intervention the asset health is revived to different degrees and therefore adds an external element of uncertainty. If a fleet comprises assets that undergo maintenance interventions, each segment of asset operations between the maintenance operations must be treated independently.

10. Collaborative prognosis relies on clustering the assets based on their operating conditions, such that the clusters of assets in homogeneous operations can be identified. Clustering therefore forms a critical segment of the collaborative prognosis pipeline. Statistical clustering algorithms rely on the asset condition data, that in turn reflect the asset operating conditions. For the industries this highlights the importance of data, both quantitatively and qualitatively, as it is. From an algorithm's perspective, any failure corresponds to the deviation in sensor data. On the other hand, expert knowledge can also be relied upon to identify such clusters.

References

[1] Al-Dahidi, S., Di Maio, F., Baraldi, P., Zio, E., and Seraoui, R. (2018). A framework for reconciliating data clusters from a fleet of nuclear power plants turbines for fault diagnosis. *Applied Soft Computing Journal*, 69:213–231.

[2] Amaral, L., Alderliesten, R., and Benedictus, R. (2018). Towards a physics-based relationship for crack growth under different loading modes. *Engineering Fracture Mechanics*, 195:222–241.

[3] Amezquita-Sanchez, J. P., Park, H. S., and Adeli, H. (2017). A novel methodology for modal parameters identification of large smart structures using music, empirical wavelet transform, and hilbert transform. *Engineering Structures*, 147:148–159.

[4] Anantharaman, M., Khan, F., Garaniya, V., and Lewarn, B. (2018). Reliability assessment of main engine subsystems considering turbocharger failure as a case study. *TransNav*, 12(2):271–276.

[5] Ansari, F., Glawar, R., and Nemeth, T. (2019). Prima: a prescriptive maintenance model for cyber-physical production systems. *International Journal of Computer Integrated Manufacturing*, 32(4-5):482–503.

[6] Atzori, L., Iera, A., and Morabito, G. (2014). From" smart objects" to" social objects": The next evolutionary step of the internet of things. *IEEE Communications Magazine*, 52(1):97–105.

[7] Bakliwal, K., Dhada, M. H., Palau, A. S., Parlikad, A. K., and Lad, B. K. (2018). A Multi Agent System architecture to implement Collaborative Learning for social industrial assets. *IFAC-PapersOnLine*, 51(11):1237–1242.

[8] Bekkerman, R., Bilenko, M., and Langford, J. (2011). *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press, United Kingdom.

[9] Bhaskar, S., Saha, S., and Goebel, K. (2009). A distributed prognostic health management architecture. Technical report.

[10] Bhattacharyya, A. (1946). On a measure of divergence between two multinomial populations. *Sankhyā: the indian journal of statistics*, pages 401–406.

[11] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, B., et al. (2019a). Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, 1:374–388.

[12] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, H. B., Van Overveldt, T., Petrou, D., Ramage, D., and Roselander, J. (2019b). Towards Federated Learning at Scale: System Design. *arXiv preprint arXiv:1902.01046*.

[13] Bonilla, E. V., Chai, K., and Williams, C. (2007). Multi-task gaussian process prediction. *Advances in neural information processing systems*, 20.

[14] Borguet, S. and Léonard, O. (2009). A generalized likelihood ratio test for adaptive gas turbine performance monitoring. *Journal of Engineering for Gas Turbines and Power*, 131(1).

[15] Brauer, D. C. and Brauer, G. D. (1987). Reliability-centered maintenance. *IEEE transactions on reliability*, 36(1):17–24.

[16] Breidbach, C. F. and Maglio, P. P. (2016). Technology-enabled value co-creation: An empirical analysis of actors, resources, and practices. *Industrial Marketing Management*, 56:73–85.

[17] Brennan, R. W., Fletcher, M., and Norrie, D. H. (2002). An agent-based approach to reconfiguration of real-time distributed control systems. *IEEE transactions on Robotics and Automation*, 18(4):444–451.

[18] Brisimi, T. S., Chen, R., Mela, T., Olshevsky, A., Paschalidis, I. C., and Shi, W. (2018). Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67.

[19] Buchanan, B. G. (1986). Expert systems: working systems and the research literature. *Expert systems*, 3(1):32–50.

[20] Bull, L., Dhada, M., Steinert, O., Lindgren, T., Parlikad, A. K., Duncan, A. B., and Girolami, M. (2022). Knowledge transfer in engineering fleets: Hierarchical bayesian modelling for multi-task learning. *arXiv preprint arXiv:2204.12404*.

[21] Bull, L., Gardner, P., Dervilis, N., Papatheou, E., Haywood-Alexander, M., Mills, R., and Worden, K. (2021a). On the transfer of damage detectors between structures: An experimental case study. *Journal of Sound and Vibration*, 501:116072.

[22] Bull, L., Gardner, P., Gosliga, J., Rogers, T., Dervilis, N., Cross, E., Papatheou, E., Maguire, A., Campos, C., and Worden, K. (2021b). Foundations of population-based SHM, part I: Homogeneous populations and forms. *Mechanical Systems and Signal Processing*, 148:107141.

[23] Bull, L., Rogers, T., Wickramarachchi, C., Cross, E., Worden, K., and Dervilis, N. (2019). Probabilistic active learning: An online framework for structural health monitoring. *Mechanical Systems and Signal Processing*, 134:106294.

[24] Cannarile, F., Compare, M., Di Maio, F., and Zio, E. (2018). A clustering approach for mining reliability big data for asset management. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 232(2):140–150.

[25] Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, 76(1).

[26] Chen, Z. et al. (2003). Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69.

[27] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to algorithms*. MIT press.

[28] Cox, D. (1983). Point processes and renewal theory: a brief survey. *Electronic systems effectiveness and life cycle costing*, pages 107–112.

[29] Cox, D. R. and Lewis, P. A. (1966). The statistical analysis of series of events.

[30] Dhada, M., Girolami, M., and Parlikad, A. K. (2020a). Anomaly detection in a fleet of industrial assets with hierarchical statistical modeling. *Data-Centric Engineering*, 1.

[31] Dhada, M., Hadjidemetriou, G. M., and Parlikad, A. K. (2020b). Predicting bridge elements deterioration, using collaborative gaussian process regression. *IFAC-PapersOnLine*, 53(3):348–353.

[32] Dhada, M., Hernández, M. P., Palau, A. S., and Parlikad, A. K. (2021). Comparison of agent deployment strategies for collaborative prognosis. In *2021 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–8. IEEE.

[33] Dhada, M., Jain, A. K., Herrera, M., Hernandez, M. P., and Parlikad, A. K. (2020c). Secure and communications-efficient collaborative prognosis. *IET Collaborative Intelligent Manufacturing*, 2(4):164–173.

[34] Dhada, M., Jain, A. K., and Parlikad, A. K. (2020d). Empirical convergence analysis of federated averaging for failure prognosis. *IFAC-PapersOnLine*, 53(3):360–365.

[35] Dhada, M. H., Palau, A. S., and Parlikad, A. K. (2019). Federated Learning for Collaborative Prognosis. In *International Conference on Precision, Meso, Micro, and Nano Engineering*, IIT Indore, India.

[36] Di Francesco, D., Chryssanthopoulos, M., Faber, M. H., and Bharadwaj, U. (2021). Decision-theoretic inspection planning using imperfect and incomplete data. *Data-Centric Engineering*, 2.

[37] Diday, E. and Noirhomme-Fraiture, M. (2008). *Symbolic data analysis and the SODAS software*. John Wiley & Sons.

[38] Diez-Olivan, A., Del Ser, J., Galar, D., and Sierra, B. (2019). Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0. *Information Fusion*, 50:92–111.

[39] Don Ranasinghe, G. (2021). *Prognostics Under the Conditions of Limited Failure Data Availability*. PhD thesis, University of Cambridge.

[40] Dorafshan, S., Thomas, R. J., and Maguire, M. (2018). Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, 186:1031–1045.

[41] Duffie, N. A. and Piper, R. S. (1986). Nonhierarchical control of manufacturing systems. *Journal of Manufacturing Systems*, 5(2):141.

[42] Economou, T., Kapelan, Z., and Bailey, T. (2007). An aggregated hierarchical bayesian model for the prediction of pipe failures. In *Proceedings of the 9th International Conference on Computing and Control for the Water Industry (CCWI), Leicester, UK*.

[43] Eker, O. F., Camci, F., and Jennions, I. K. (2014). A Similarity-based Prognostics Approach for Remaining Useful Life Prediction. In *Second European Conference of the Prognostics and Health Management Society, Nantes, France*. PHM Society.

[44] Evans, P. C. and Annunziata, M. (2012). Industrial internet: Pushing the boundaries. *General Electric Reports*.

[45] Frisk, E., Krysander, M., and Larsson, E. (2014). Data-driven lead-acid battery prognostics using random survival forests. Technical report, Linkoping University Linkoping Sweden.

[46] Gao, Y. and Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, 33(9):748–768.

[47] Gardner, P., Bull, L., Dervilis, N., and Worden, K. (2021a). Overcoming the problem of repair in structural health monitoring: Metric-informed transfer learning. *Journal of Sound and Vibration*, page 116245.

[48] Gardner, P., Bull, L., Dervilis, N., and Worden, K. (2022). On the application of kernelised bayesian transfer learning to population-based structural health monitoring. *Mechanical Systems and Signal Processing*, 167:108519.

[49] Gardner, P., Bull, L., Gosliga, J., Dervilis, N., and Worden, K. (2021b). Foundations of population-based SHM, part III: Heterogeneous populations–mapping and transfer. *Mechanical Systems and Signal Processing*, 149:107142.

[50] Gardner, P., Fuentes, R., Dervilis, N., Mineo, C., Pierce, S., Cross, E., and Worden, K. (2020a). Machine learning at the interface of structural health monitoring and non-destructive evaluation. *Philosophical Transactions of the Royal Society A*, 378(2182):20190581.

[51] Gardner, P., Liu, X., and Worden, K. (2020b). On the application of domain adaptation in structural health monitoring. *Mechanical Systems and Signal Processing*, 138:106550.

[52] Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC, third edition.

[53] Gelman, A. and Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.

[54] Gilchrist, A. (2016). Iiot reference architecture. In *Industry 4.0*, pages 65–86. Springer.

[55] Giret, A. and Botti, V. (2004). Holons and agents. *Journal of intelligent manufacturing*, 15(5):645–659.

[56] Goebel, K., Saha, B., Saxena, A., Mct, N., and Riacs, N. (2008). A comparison of three data-driven techniques for prognostics. In *62nd meeting of the society for machinery failure prevention technology (mfpt)*, pages 119–131.

[57] González-Prida, V., Orchard, M., Martín, C., Guillén, A., Shambhu, J., and Shariff, S. (2016). Case Study based on Inequality Indices for the Assessments of Industrial Fleets. *IFAC-PapersOnLine*, 49(28):250–255.

[58] González-Prida, V., Parra, C., Márquez, A. C., Pérès, F., and Loren, C. M. (2021). Practical implementation of an asset management system according to iso 55001. a future direction in the cloud & iot paradigm. Technical report, EasyChair.

[59] Gosliga, J., Gardner, P., Bull, L., Dervilis, N., and Worden, K. (2021). Foundations of population-based SHM, part II: Heterogeneous populations–graphs, networks, and communities. *Mechanical Systems and Signal Processing*, 148:107144.

[60] Goyal, D. and Pabla, B. S. (2015). Condition based maintenance of machine tools-A review. *CIRP Journal of Manufacturing Science and Technology*, 10:24–35.

[61] Grubic, T. (2018). Remote monitoring technology and servitization: Exploring the relationship. *Computers in Industry*, 100:148–158.

[62] Gurung, R., Lindgren, T., and Boström, H. (2018). Learning random forest from histogram data using split specific axis rotation. *International Journal of Machine Learning and Computing*, 8(1):74–79.

[63] Gurung, R. B., Lindgren, T., Bostr, H., et al. (2017). Predicting nox sensor failure in heavy duty trucks using histogram-based random forests. *International Journal of Prognostics and Health Management*, 8(1).

[64] Gurung, R. B., Lindgren, T., and Boström, H. (2016). Learning decision trees from histogram data using multiple subsets of bins. In *Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, FLAIRS, Key Largo, Florida, May 16-18, 2016*, pages 430–435. AAAI Press.

[65] Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. (2018). Federated Learning for Mobile Keyboard Prediction. *arXiv preprint arXiv:1811.03604*.

[66] Hashemian, H. M. (2010). State-of-the-art predictive maintenance techniques. *IEEE Transactions on Instrumentation and measurement*, 60(1):226–236.

[67] Hastings, N. A. et al. (2010). *Physical asset management*, volume 2. Springer.

[68] He, N., Zhang, D., and Li, Q. (2014). Agent-based hierarchical production planning and scheduling in make-to-order manufacturing system. *International Journal of Production Economics*, 149:117–130.

[69] Hernández, J. E., Lyons, A. C., Mula, J., Poler, R., and Ismail, H. (2014). Supporting the collaborative decision-making process in an automotive supply chain with a multi-agent system. *Production Planning & Control*, 25(8):662–678.

[70] Hernández, M. P., Mcfarlane, D., Parlikad, A. K., Herrera, M., and Jain, A. K. (2021). Relaxing platform dependencies in agent-based control systems. *IEEE Access*, 9:30511–30527.

[71] Herrera, M., Izquierdo, J., Pérez-García, R., and Ayala-Cabrera, D. (2010). Water supply clusters by multi-agent based approach. In *Water distribution systems analysis 2010*, pages 861–869.

[72] Herrera, M., Izquierdo, J., Pérez-García, R., and Montalvo, I. (2012). Multi-agent adaptive boosting on semi-supervised water supply clusters. *Advances in Engineering Software*, 50:131–136.

[73] Herrera, M., Pérez-Hernández, M., Kumar Parlikad, A., and Izquierdo, J. (2020). Multi-agent systems and complex networks: Review and applications in systems engineering. *Processes*, 8(3):312.

[74] Hidalgo-Mompeán, F., Fernández, J. F. G., Cerruela-García, G., and Márquez, A. C. (2021). Dimensionality analysis in machine learning failure detection models. a case study with lng compressors. *Computers in Industry*, 128:103434.

[75] Hoffman, M. D., Gelman, A., et al. (2014). The No-U-Turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623.

[76] Huang, L., Yin, Y., Fu, Z., Zhang, S., Deng, H., and Liu, D. (2018). LoAdaBoost:Loss-Based AdaBoost Federated Machine Learning on medical Data. *arXiv preprint arXiv:1811.12629*.

[77] Huang, Y. and Beck, J. L. (2015). Hierarchical sparse Bayesian learning for strucutral health monitoring with incomplete modal data. *International Journal for Uncertainty Quantification*, 5(2).

[78] Huang, Y., Beck, J. L., and Li, H. (2019). Multitask sparse Bayesian learning with applications in structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, 34(9):732–754.

[79] Izquierdo, J., Herrera, M., Montalvo, I., and Pérez-García, R. (2009). Division of water supply systems into district metered areas using a multi-agent based approach. In *International Conference on Software and Data Technologies*, pages 167–180. Springer.

[80] Jain, A. K. and Lad, B. K. (2016). Data driven models for prognostics of high speed milling cutters. *International Journal of Performability Engineering*, 12(1):3–11.

[81] Jain, A. K. and Lad, B. K. (2017). Dynamic optimization of process quality control and maintenance planning. *IEEE Transactions on Reliability*, 66(2):502–517.

[82] Jang, K., Kim, N., and An, Y.-K. (2019). Deep learning–based autonomous concrete crack evaluation through hybrid image scanning. *Structural Health Monitoring*, 18(5-6):1722–1737.

[83] Jin, C., Djurdjanovic, D., Ardakani, H. D., Wang, K., Buzza, M., Begheri, B., Brown, P., and Lee, J. (2015). A comprehensive framework of factory-to-factory dynamic fleet-level prognostics and operation management for geographically distributed assets. In *IEEE International Conference on Automation Science and Engineering*, volume 2015-Octob, pages 225–230. IEEE Computer Society.

[84] Jin, X., Ma, E. W., Cheng, L. L., and Pecht, M. (2012). Health monitoring of cooling fans based on mahalanobis distance with mRMR feature selection. *IEEE Transactions on Instrumentation and Measurement*, 61(8):2222–2229.

[85] Jindal, K., Srinivasan, S., and Sharma, M. (2013). Review of decision support system based on multi agent in production scheduling. *International Journal of Engineering and Social Science*, 3(10):33–36.

[86] Jochems, A., Deist, T. M., Van Soest, J., Eble, M., Bulens, P., Coucke, P., Dries, W., Lambin, P., and Dekker, A. (2016). Distributed learning: developing a predictive model based on data from multiple hospitals without data leaving the hospital–a real life proof of concept. *Radiotherapy and Oncology*, 121(3):459–467.

[87] Johnson, V. E., Moosman, A., and Cotter, P. (2005). A hierarchical model for estimating the early reliability of complex systems. *IEEE Transactions on Reliability*, 54(2):224–231.

[88] Kang, M. (2018). Machine Learning: Anomaly Detection. In *Prognostics and Health Management of Electronics*, pages 131–162. John Wiley and Sons Ltd, Chichester, UK.

[89] Kao, L.-J. and Chen, H.-F. (2012). Applying hierarchical bayesian neural network in failure time prediction. *Mathematical Problems in Engineering*, 2012.

[90] Khan, S. S. and Madden, M. G. (2010). A survey of recent trends in one class classification. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6206 LNAI, pages 188–197. Springer, Berlin, Heidelberg.

[91] Kimotho, J. K. and Sextro, W. (2014). An approach for feature extraction and selection from non-trending data for machinery prognosis. In *PHM Society European Conference*, volume 2.

[92] Kobayashi, T. and Simon, D. L. (2005). Evaluation of an enhanced bank of Kalman filters for in-flight aircraft engine sensor fault diagnostics. *Journal of Engineering for Gas Turbines and Power*, 127(3):497–504.

[93] Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

[94] Kreft, I. G. and De Leeuw, J. (1998). *Introducing Multilevel Modeling*. Sage.

[95] Kusiak, A. (2018). Smart manufacturing. *International Journal of Production Research*, 56(1-2):508–517.

[96] Kusiak, A. and Li, W. (2011). The prediction and diagnosis of wind turbine faults. *Renewable energy*, 36(1):16–23.

[97] Lapira, E. R. and Lee, J. (2012). *Fault detection in a network of similar machines using clustering approach*. PhD thesis, University of Cincinnati.

[98] Lee, J., Jin, C., Liu, Z., and Ardakani, H. D. (2017). Introduction to data-driven methodologies for prognostics and health management. In Ekwaro-Osire, S., Carlos Gonçalves, A., and M. Alemayehu, F., editors, *Probabilistic Prognostics and Health Management of Energy Systems*, pages 9–32. Springer, Switzerland.

[99] 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(1-2):314–334.

[100] Leitão, P. and Karnouskos, S. (2015). *Industrial Agents: Emerging Applications of Software Agents in Industry*. Morgan Kaufmann.

[101] Leone, G., Cristaldi, L., and Turrin, S. (2016). A data-driven prognostic approach based on sub-fleet knowledge extraction. In *14th IMEKO TC10 Workshop on Technical Diagnostics: New Perspectives in Measurements, Tools and Techniques for Systems Reliability, Maintainability and Safety*, pages 417–422.

[102] Leone, G., Cristaldi, L., and Turrin, S. (2017). A data-driven prognostic approach based on statistical similarity: An application to industrial circuit breakers. *Measurement: Journal of the International Measurement Confederation*, 108:163–170.

[103] Lester, D. (2015). Driving servitization: David lester.

[104] Li, H., Palau, A. S., and Parlikad, A. K. (2018a). A social network of collaborating industrial assets. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 232(4):389–400.

[105] Li, H. and Parlikad, A. K. (2017). Study of dynamic workload assignment strategies on production performance. *IFAC-PapersOnLine*, 50(1):13710–13715.

[106] Li, L., Fan, Y., Tse, M., and Lin, K.-Y. (2020). A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854.

[107] Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2019a). Federated Learning: Challenges, Methods, and Future Directions. *arXiv preprint arXiv:1908.07873*.

[108] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2018b). Federated Optimization in Heterogeneous Networks. *arXiv preprint arXiv:1812.06127*.

[109] Li, X., Zhang, W., Ding, Q., and Sun, J.-Q. (2019b). Multi-layer domain adaptation method for rolling bearing fault diagnosis. *Signal Processing*, 157:180–197.

[110] Li, Y., Bao, T., Chen, Z., Gao, Z., Shu, X., and Zhang, K. (2021). A missing sensor measurement data reconstruction framework powered by multi-task gaussian process regression for dam structural health monitoring systems. *Measurement*, 186:110085.

[111] Li, Z., Park, H. S., and Adeli, H. (2017). New method for modal identification of super high-rise building structures using discretized synchrosqueezed wavelet and hilbert transforms. *The Structural Design of Tall and Special Buildings*, 26(3):e1312.

[112] Liang, Z. and Parlikad, A. K. (2020). Predictive group maintenance for multi-system multi-component networks. *Reliability Engineering & System Safety*, 195:106704.

[113] Lin, Y., Liu, S., and Huang, S. (2018). Selective sensing of a heterogeneous population of units with dynamic health conditions. *IISE Transactions*, 50(12):1076–1088.

[114] Liu, J. and Zio, E. (2016). A framework for asset prognostics from fleet data. In *2016 prognostics and system health management conference (phm-chengdu)*, pages 1–5. IEEE.

[115] Liu, Z. (2018). *Cyber-Physical System Augmented Prognostics and Health Management for Fleet-Based Systems*. PhD thesis, University of Cincinnati.

[116] Macchi, M., Roda, I., Negri, E., and Fumagalli, L. (2018). Exploring the role of digital twin for asset lifecycle management. *IFAC-PapersOnLine*, 51(11):790–795.

[117] Madhusudana, C., Kumar, H., and Narendranath, S. (2016). Condition monitoring of face milling tool using k-star algorithm and histogram features of vibration signal. *Engineering science and technology, an international journal*, 19(3):1543–1551.

[118] Mařík, V. and Lažanskỳ, J. (2007). Industrial applications of agent technologies. *Control Engineering Practice*, 15(11):1364–1380.

[119] Martinez, V., Neely, A., Velu, C., Leinster-Evans, S., and Bisessar, D. (2017). Exploring the journey to services. *International Journal of Production Economics*, 192:66–80.

[120] Mcfarlane, D. (2019). Industrial Internet of Things Applying IoT in the Industrial Context. Technical report, Institute for Manufacturing, University of Cambridge.

[121] Michau, G. and Fink, O. (2019a). Domain Adaptation for One-Class Classification: Monitoring the Health of Critical Systems Under Limited Information. *http://arxiv.org/abs/1907.09204*.

[122] Michau, G. and Fink, O. (2019b). Domain adaptation for one-class classification: monitoring the health of critical systems under limited information. *arXiv preprint arXiv:1907.09204*.

[123] Michau, G., Palmé, T., and Fink, O. (2018a). Fleet PHM for Critical Systems: Bi-level Deep Learning Approach for Fault Detection. In *Proceedings of the European Conference of the PHM Society*.

[124] Michau, G., Palmé, T., and Fink, O. (2018b). Fleet phm for critical systems: bi-level deep learning approach for fault detection. In *Proceedings of the European Conference of the PHM Society 2018*, volume 4, page 403. PHM Society.

[125] Monostori, L., Váncza, J., and Kumara, S. R. (2006). Agent-based systems for manufacturing. *CIRP Annals-Manufacturing Technology*, 55(2):697–720.

[126] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT press.

[127] Nascimento, R. G. and Viana, F. A. (2019). Fleet prognosis with physics-informed recurrent neural networks. *arXiv preprint arXiv:1901.05512*.

[128] Neal, R. M. (2000). Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265.

[129] Neely, A. (2008). Exploring the financial consequences of the servitization of manufacturing. *Operations Management Research*, 1(2):103–118.

[130] Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., Niyato, D., and Poor, H. V. (2021). Federated learning for industrial internet of things in future industries. *IEEE Wireless Communications*.

[131] Ning, H. and Wang, Z. (2011). Future internet of things architecture: like mankind neural system or social organization framework? *IEEE Communications Letters*, 15(4):461–463.

[132] Nwana, H. S. (1996). Software agents: An overview. *The knowledge engineering review*, 11(3):205–244.

[133] O'Connor, P. and Kleyner, A. (2012). *Practical Reliability Engineering*. John Wiley & Sons.

[134] Orchard, M. E., Tang, L., and Vachtsevanos, G. (2011). A combined anomaly detection and failure prognosis approach for estimation of remaining useful life in energy storage devices. In *Annual Conference of the PHM Society*, volume 3.

[135] Palau, A. S., Bakliwal, K., Dhada, M. H., Pearce, T., and Parlikad, A. K. (2018). Recurrent Neural Networks for real-time distributed collaborative prognostics. In *2018 IEEE International Conference on Prognostics and Health Management, ICPHM 2018*. Institute of Electrical and Electronics Engineers Inc.

[136] Palau, A. S., Dhada, M. H., Bakliwal, K., and Parlikad, A. K. (2019a). An industrial multi agent system for real-time distributed collaborative prognostics. *Engineering Applications of Artificial Intelligence*, 85:590–606.

[137] Palau, A. S., Dhada, M. H., and Parlikad, A. K. (2019b). Multi-agent system architectures for collaborative prognostics. *Journal of Intelligent Manufacturing*, pages 1–15.

[138] Palau, A. S., Liang, Z., Lütgehetmann, D., and Parlikad, A. K. (2019c). Collaborative prognostics in social asset networks. *Future Generation Computer Systems*, 92:987–995.

[139] Paleyes, A., Urma, R.-G., and Lawrence, N. D. (2020). Challenges in deploying machine learning: a survey of case studies. *arXiv preprint arXiv:2011.09926*.

[140] Papadimas, N. and Dodwell, T. (2021). A hierarchical Bayesian approach for calibration of stochastic material models. *Data-Centric Engineering*, 2.

[141] Pastore, M. and Calcagnì, A. (2019). Measuring distribution similarities between samples: A distribution-free overlapping index. *Frontiers in psychology*, 10:1089.

[142] Patrick, R., Smith, M. J., Byington, C. S., Vachtsevanos, G. J., Tom, K., and Ly, C. (2010). Integrated software platform for fleet data analysis, enhanced diagnostics, and safe transition to prognostics for helicopter component cbm. Technical report, ARMY RESEARCH LAB ADELPHI MD.

[143] Perez-Ramirez, C. A., Amezquita-Sanchez, J. P., Valtierra-Rodriguez, M., Adeli, H., Dominguez-Gonzalez, A., and Romero-Troncoso, R. J. (2019). Recurrent neural network model with bayesian training and mutual information for response prediction of large buildings. *Engineering Structures*, 178:603–615.

[144] Perreault, D. J. and Caliskan, V. (2000). A new design for automotive alternators. In *SAE CONFERENCE PROCEEDINGS P*, pages 583–594. Citeseer.

[145] Petchrompo, S. and Parlikad, A. K. (2019). A review of asset management literature on multi-asset systems. *Reliability Engineering & System Safety*, 181:181–201.

[146] Pham, Q.-V., Dev, K., Maddikunta, P. K. R., Gadekallu, T. R., Huynh-The, T., et al. (2021). Fusion of federated learning and industrial internet of things: A survey. *arXiv preprint arXiv:2101.00798*.

[147] Plaksin, A., Gritsenko, A., and Glemba, K. (2015). Modernization of the turbocharger lubrication system of an internal combustion engine. *Procedia Engineering*, 129:857–862.

[148] Prytz, R., Nowaczyk, S., Rögnvaldsson, T., and Byttner, S. (2015). Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. *Engineering applications of artificial intelligence*, 41:139–150.

[149] Puzakov, A. (2019). Physical modeling of failures of the automotive alternator. In *IOP Conference Series: Materials Science and Engineering*, volume 643, page 012019. IOP Publishing.

[150] Rahat, M., Pashami, S., Nowaczyk, S., and Kharazian, Z. (2020). Modeling turbocharger failures using markov process for predictive maintenance. In *30th European Safety and Reliability Conference (ESREL2020) & 15th Probabilistic Safety Assessment and Management Conference (PSAM15), Venice, Italy, 1-5 November, 2020*. European Safety and Reliability Association.

[151] Rajabzadeh, Y., Rezaie, A. H., and Amindavar, H. (2016). A dynamic modeling approach for anomaly detection using stochastic differential equations. *Digital Signal Processing: A Review Journal*, 54:1–11.

[152] Ranasinghe, G. D., Lindgren, T., Girolami, M., and Parlikad, A. K. (2019). A Methodology for Prognostics under the Conditions of Limited Failure Data Availability. *IEEE Access*, 7:183996–184007.

[153] Reymonet, A., Thomas, J., and Aussenac-Gilles, N. (2009). Ontology based information retrieval: an application to automotive diagnosis. In *International Workshop on Principles of Diagnosis (DX 2009)*, pages 9–14. Linköping University, Institut of Technology.

[154] Sadeghi, A. R., Wachsmann, C., and Waidner, M. (2015). Security and privacy challenges in industrial Internet of Things. In *Proceedings of the 52nd Annual Design Automation Conference, San Francisco, USA*. Institute of Electrical and Electronics Engineers Inc.

[155] Salvador Palau, A. (2020). *Distributed collaborative prognostics*. PhD thesis, University of Cambridge.

[156] Salvador Palau, A., Dhada, M. H., Bakliwal, K., and Parlikad, A. K. (2019a). An Industrial Multi Agent System for real-time distributed collaborative prognostics. *Engineering Applications of Artificial Intelligence*, 85:590–606.

[157] Salvador Palau, A., Dhada, M. H., and Parlikad, A. K. (2019b). Multi-agent system architectures for collaborative prognostics. *Journal of Intelligent Manufacturing*, 30(8):2999–3013.

[158] Salvador Palau, A., Liang, Z., Lütgehetmann, D., and Parlikad, A. K. (2019c). Collaborative prognostics in Social Asset Networks. *Future Generation Computer Systems*, 92:987–995.

[159] Saxena, A. and Goebel, K. (2008). Turbofan Engine Degradation Simulation Data Set. Technical report, NASA Ames Research Center, Moffett Field, CA.

[160] Saxena, A., Goebel, K., Simon, D., and Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *International Conference on Prognostics and Health Management, Denver, USA*.

[161] Schwabacher, M. and Goebel, K. (2007). A Survey of Artificial Intelligence for Prognostics. In *AAAI Fall Symposium: Artificial Intelligence for Prognostics, Virginia, USA*.

[162] Seshadri, P., Duncan, A., Thorne, G., Parks, G., Dıaz, R. V., and Girolami, M. (2020). Bayesian assessments of aeroengine performance with transfer learning. *arXiv preprint arXiv:2011.14698*.

[163] Sharma, A., Kosasih, E., Zhang, J., Brintrup, A., and Calinescu, A. (2020). Digital twins: State of the art theory and practice, challenges, and open research questions. *arXiv preprint arXiv:2011.02833*.

[164] Shen, W., Hao, Q., Yoon, H. J., and Norrie, D. H. (2006). Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced engineering INFORMATICS*, 20(4):415–431.

[165] Siemieniuch, C. E. and Sinclair, M. A. (2002). On complexity, process ownership and organisational learning in manufacturing organisations, from an ergonomics perspective. *Applied Ergonomics*, 33(5):449–462.

[166] Smith, D. J. (2013). Power-by-the-hour: the role of technology in reshaping business strategy at rolls-royce. *Technology analysis & strategic management*, 25(8):987–1007.

[167] Spiewak, S., Duggirala, R., and Barnett, K. (2000). Predictive monitoring and control of the cold extrusion process. *CIRP Annals*, 49(1):383–386.

[168] Stan Development Team (2021). RStan: the R interface to Stan. R package version 2.21.3.

[169] Sukhija, S. and Krishnan, N. C. (2020). Shallow domain adaptation. In *Domain Adaptation in Computer Vision with Deep Learning*, pages 23–40. Springer.

[170] Sun, Z., Barp, A., and Briol, F.-X. (2021). Vector-valued control variates. *arXiv preprint arXiv:2109.08944*.

[171] Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337.

[172] Tan-Kim, A., Hagen, N., Lanfranchi, V., Clénet, S., Coorevits, T., Mipo, J.-C., Legranger, J., and Palleschi, F. (2017). Influence of the manufacturing process of a claw-pole alternator on its stator shape and acoustic noise. *IEEE Transactions on Industry Applications*, 53(5):4389–4395.

[173] Teoh, Y. K., Gill, S. S., and Parlikad, A. K. (2021). Iot and fog computing based predictive maintenance model for effective asset management in industry 4.0 using machine learning. *IEEE Internet of Things Journal*.

[174] Thawkar, A., Tambe, P., and Deshpande, V. (2018). A reliability centred maintenance approach for assessing the impact of maintenance for availability improvement of carding machine. *International Journal of Process Management and Benchmarking*, 8(3):318–339.

[175] Thill, M. (2017). The Relationship between the Mahalanobis Distance and the Chi-Squared Distribution.

[176] Tjiparuro, Z. and Thompson, G. (2004). Review of maintainability design principles and their application to conceptual design. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, 218(2):103–113.

[177] Tsialiamanis, G., Mylonas, C., Chatzi, E., Wagg, D., Dervilis, N., and Worden, K. (2022). On an application of graph neural networks in population-based shm. In *Data Science in Engineering, Volume 9*, pages 47–63. Springer.

[178] Turconi, G., Ventola, G., González-Prida, V., Parra, C., and Crespo, A. (2022). A literature review on lean manufacturing in the industry 4.0: From integrated systems to iot and smart factories. *IoT and Cloud Computing for Societal Good*, pages 181–194.

[179] Uçar, M., Bayir, R., and Özer, M. (2009). Real time detection of alternator failures using intelligent control systems. In *2009 International Conference on Electrical and Electronics Engineering-ELECO 2009*, pages II–380. IEEE.

[180] Upasani, K., Bakshi, M., Pandhare, V., and Lad, B. K. (2017). Distributed maintenance planning in manufacturing industries. *Computers & Industrial Engineering*, 108:1–14.

[181] Vachtsevanos, G. J. and Vachtsevanos, G. J. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*, volume 456. Wiley Online Library.

[182] Vandermerwe, S. and Rada, J. (1988). Servitization of business: adding value by adding services. *European management journal*, 6(4):314–324.

[183] Vrba, P. (2013). Review of industrial applications of multi-agent technologies. In *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics*, pages 327–338. Springer.

[184] Wan, H.-P. and Ni, Y.-Q. (2019). Bayesian multi-task learning methodology for reconstruction of structural health monitoring data. *Structural Health Monitoring*, 18(4):1282–1309.

[185] Wand, M. (2009). Semiparametric regression and graphical models. *Australian & New Zealand Journal of Statistics*, 51(1):9–41.

[186] Wang, Q., Michau, G., and Fink, O. (2019). Domain adaptive transfer learning for fault diagnosis. In *2019 Prognostics and System Health Management Conference (PHM-Paris)*, pages 279–285. IEEE.

[187] Wang, S., Wan, J., Zhang, D., Li, D., and Zhang, C. (2016). Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, 101:158–168.

[188] Wang, T., Yu, J., Siegel, D., and Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *2008 International Conference on Prognostics and Health Management, Denver, USA*.

[189] Watson, N. and Janota, M. (1982). *Turbocharging the internal combustion engine*. Macmillan International Higher Education.

[190] West, B. T., Welch, K. B., and Galecki, A. T. (2006). *Linear Mixed Models: A Practical Guide Using Statistical Software*. Chapman and Hall/CRC.

[191] Wong, T., Leung, C., Mak, K.-L., and Fung, R. Y. (2006). Dynamic shopfloor scheduling in multi-agent manufacturing systems. *Expert Systems with Applications*, 31(3):486–494.

[192] Wu, D., Jennings, C., Terpenny, J., Gao, R. X., and Kumara, S. (2017). A comparative study on machine learning algorithms for smart manufacturing: tool wear prediction using random forests. *Journal of Manufacturing Science and Engineering*, 139(7).

[193] Xiang, W. and Lee, H. P. (2008). Ant colony intelligence in multi-agent dynamic manufacturing scheduling. *Engineering Applications of Artificial Intelligence*, 21(1):73–85.

[194] Xu, L., Mak, S., and Brintrup, A. (2021). Will bots take over the supply chain? revisiting agent-based supply chain automation. *International Journal of Production Economics*, 241:108279.

[195] Yan, W. (2016). One-class extreme learning machines for gas turbine combustor anomaly detection. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2016-Octob, pages 2909–2914. Institute of Electrical and Electronics Engineers Inc.

[196] Yang, Q., Liu, Y., Chen, T., and Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19.

[197] Yucesan, Y. A., Dourado, A., and Viana, F. A. (2021). A survey of modeling for prognosis and health management of industrial equipment. *Advanced Engineering Informatics*, 50:101404.

[198] Zaccaria, V., Stenfelt, M., Aslanidou, I., and Kyprianidis, K. G. (2018). Fleet monitoring and diagnostics framework based on digital twin of aero-engines. In *Turbo Expo: Power for Land, Sea, and Air*, volume 51128, page V006T05A021. American Society of Mechanical Engineers.

[199] Zaidan, M. A., Harrison, R. F., Mills, A. R., and Fleming, P. J. (2015). Bayesian Hierarchical Models for aerospace gas turbine engine prognostics. *Expert Systems with Applications*, 42(1):539–553.

[200] Zhang, B., Hong, X., and Liu, Y. (2020). Multi-task deep transfer learning method for guided wave-based integrated health monitoring using piezoelectric transducers. *IEEE Sensors Journal*, 20(23):14391–14400.

[201] Zhang, P., Wang, C., Jiang, C., and Han, Z. (2021). Deep reinforcement learning assisted federated learning algorithm for data management of iiot. *IEEE Transactions on Industrial Informatics*, 17(12):8475–8484.

[202] Zhang, W., Peng, G., Li, C., Chen, Y., and Zhang, Z. (2017). A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors*, 17(2):425.

[203] Zio, E. and Di Maio, F. (2010). A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. *Reliability Engineering and System Safety*, 95(1):49–57.

Appendix A

Derivations of the E and M steps

E-step

For the case of asset fleets, the E-step involves first evaluating the expectation of $\mathbf{z}$ w.r.t. distribution conditioned on $\mathbf{X}$ for parameter values $\theta = \theta^t$. Since $\mathbf{z}_{i,k}$ is binary, $\mathbb{E}(\mathbf{z}_{i,k}|(\mu_i, \mathbf{C}_i)^t, \theta^t) = p(\mathbf{z}_{i,k} = 1|(\mu_i, \mathbf{C}_i)^t, \theta^t) = p(\mathbf{z}_{i,k} = 1|(\mu_i, \mathbf{C}_i)^t, \theta^t)$. Using Bayes' rule:

$$p(\mathbf{z}_{i,k} = 1|(\mu_i, \mathbf{C}_i)^t, \theta^t) = \frac{p((\mu_i, \mathbf{C}_i)^t|\mathbf{z}_{i,k} = 1, \theta^t)p(\mathbf{z}_{i,k} = 1)}{\sum_{k=1}^{K} p((\mu_i, \mathbf{C}_i)^t|\mathbf{z}_{i,k} = 1, \theta^t)p(\mathbf{z}_{i,k} = 1)} \tag{A.1}$$

from equations 3.8 and 3.10 we know,

$$p(\mathbf{z}_{i,k} = 1|(\mu_i, \mathbf{C}_i)^t, \theta^t) = \frac{(\mathcal{N}(\mu_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\Lambda_k, \alpha_k))(\pi_k)}{\sum_{k=1}^{K}(\mathcal{N}(\mu_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\Lambda_k, \alpha_k))(\pi_k)} \tag{A.2}$$

Where all distribution parameters correspond to the values obtained at M-step of latest (t^{th}) iteration. Let, $p(\mathbf{z}_{i,k} = 1|(\mu_i, \mathbf{C}_i)^t, \theta^t) = \gamma_{i,k}$. Therefore, our function $Q(\theta, \theta^t)$ can be deduced from equation 3.14 by replacing $\mathbf{z}_{i,k}$ with $\gamma_{i,k}$:

$$Q(\theta, \theta^t) = \sum_{i=1}^{I}\sum_{n=1}^{N_i} \log(\mathcal{N}(\mu_i, \mathbf{C}_i)) + \sum_{i=1}^{I}\sum_{k=1}^{K} \gamma_{i,k} \log\left(\pi_k \mathcal{N}(\mu_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\Lambda_k, \alpha_k)\right) \tag{A.3}$$

After substituting the symbolic representation with the corresponding distribution functions and parameters, $Q(\theta, \theta^t)$ (not including constant terms, because they would become zero after differentiation) becomes:

$$
\begin{aligned}
Q(\theta,\theta^t) = & \\
& -\frac{1}{2}\sum_i\sum_n log|\mathbf{C}_i| - \frac{1}{2}\sum_i\sum_n(\mathbf{x}_{i,n}-\mu_i)^T\mathbf{C}_i^{-1}(\mathbf{x}_{i,n}-\mu_i)- \\
& \frac{1}{2}\sum_i\gamma_{i,k}\sum_k\log|\mathbf{C}_i| + \frac{1}{2}\sum_i\gamma_{i,k}\sum_k\log(\beta_k) - \frac{1}{2}\sum_i\gamma_{i,k}\sum_k\beta_k(\mu_i-\mathbf{m}_k)^T\mathbf{C}_i^{-1}(\mu_i-\mathbf{m}_k)+ \\
& \frac{1}{2}\sum_i\gamma_{i,k}\sum_k\alpha_k\log|\Lambda_k| - \frac{1}{2}\sum_i\gamma_{i,k}\sum_k\alpha_k d\log(2)- \\
& \sum_i\gamma_{i,k}\sum_k\log\left(\Gamma_d(\frac{\alpha_k}{2})\right) - \frac{1}{2}\sum_i\gamma_{i,k}\sum_k(\alpha_k+d+1)\log|\mathbf{C}_i|- \\
& \frac{1}{2}\sum_i\gamma_{i,k}\sum_k Tr(\Lambda_k\mathbf{C}_i^{-1}) + \sum_i\gamma_{i,k}\sum_k\pi_k
\end{aligned}
\tag{A.4}
$$

The $\gamma_{i,k}$ are not included in summations because they are supposed to be treated as constants in the M-step that follows.

M-step

In M-step, θ^{t+1} values are obtained for following $(t+1)^{th}$ E-step by maximising the $Q(\theta,\theta^t)$ function obtained in equation A.4 with respect to each of the θ parameters, and treating $\gamma_{i,k}$ as constants. Calculations for partial derivatives of $Q(\theta,\theta^t)$ w.r.t. each parameter are shown below:

Evaluating $\hat{\mu}_i$

$$
\begin{aligned}
\frac{\partial Q(\theta,\theta^t)}{\partial\mu_i} &\implies \sum_n\mathbf{C}_i^{-1}(\mathbf{x}_{i,n}-\mu_i) - \sum_k\beta_k\gamma_{i,k}\mathbf{C}_i^{-1}(\mu_i-\mathbf{m}_k) = 0 \\
&\implies \sum_n\mathbf{x}_{i,n} - N_i\mu_i = \mu_i\sum_k\beta_k\gamma_{i,k} - \sum_k\beta_k\gamma_{i,k}\mathbf{m}_k \\
&\implies \hat{\mu}_i = \frac{1}{N_i+\sum_{k=1}^K\beta_k\gamma_{i,k}}\left[\sum_{n=1}^{N_i}\mathbf{x}_{i,n} + \sum_{k=1}^K\beta_k\gamma_{i,k}\mathbf{m}_k\right]
\end{aligned}
$$

Evaluating $\hat{\mathbf{m}}_k$

$$
\begin{aligned}
\frac{\partial Q(\theta,\theta^t)}{\partial \mathbf{m}_k} &\implies \sum_i \gamma_{i,k}\beta_k \mathbf{C}_i^{-1}(\mu_i - \mathbf{m}_k) = 0 \\
&\implies \beta_k \sum_i \gamma_{i,k}\mathbf{C}_i^{-1}\mu_i = \beta_k \sum_i \gamma_{i,k}\mathbf{C}_i^{-1}\mathbf{m}_k \\
&\implies \left[\sum_i \gamma_{i,k}\mathbf{C}_i^{-1}\right]\mathbf{m}_k = \left[\sum_i \gamma_{i,k}\mathbf{C}_i^{-1}\mu_i\right] \\
&\implies \hat{\mathbf{m}}_k = \left[\sum_{i=1}^{I} \gamma_{i,k}\mathbf{C}_i^{-1}\right]^{-1}\left[\sum_{i=1}^{I} \gamma_{i,k}\mathbf{C}_i^{-1}\mu_i\right]
\end{aligned}
$$

Evaluating $\hat{\Lambda}_k$

$$
\begin{aligned}
\frac{\partial Q(\theta,\theta^t)}{\partial \Lambda_k} &\implies \frac{1}{2}\sum_i \gamma_{i,k}\alpha_k \Lambda_k^{-1} - \frac{1}{2}\sum_i \gamma_{i,k}\mathbf{C}_i^{-1} = 0 \\
&\implies \Lambda_k^{-1} = \frac{\sum_i \gamma_{i,k}\mathbf{C}_i^{-1}}{\alpha_k \sum_i \gamma_{i,k}} \\
&\implies \hat{\Lambda}_k = \left[\alpha_k \sum_{i=1}^{I} \gamma_{i,k}\right]\left[\sum_{i=1}^{I} \gamma_{i,k}\mathbf{C}_i^{-1}\right]^{-1}
\end{aligned}
$$

Evaluating $\hat{\beta}_k$

$$
\begin{aligned}
\frac{\partial Q(\theta,\theta^t)}{\partial \beta_k} &\implies \frac{d}{2}\sum_i \gamma_{i,k}\frac{1}{\beta_k} - \frac{1}{2}\sum_i \gamma_{i,k}(\mu_i - \mathbf{m}_k)^T\mathbf{C}_i^{-1}(\mu_i - \mathbf{m}_k) = 0 \\
&\implies \frac{1}{\hat{\beta}_k} = \frac{\sum_{i=1}^{I} \gamma_{i,k}(\mu_i - \mathbf{m}_k)^T\mathbf{C}_i^{-1}(\mu_i - \mathbf{m}_k)}{d\sum_{i=1}^{I} \gamma_{i,k}}
\end{aligned}
$$

Evaluating $\hat{\mathbf{C}}_i$

$$
\begin{aligned}
\frac{\partial Q(\theta,\theta^t)}{\partial \mathbf{C}_i} &\implies -\frac{N_i}{2}\mathbf{C}_i^{-1} + \frac{1}{2}\mathbf{C}_i^{-1}\big(\sum_n (\mathbf{x}_{i,n}-\mu_i)(\mathbf{x}_{i,n}-\mu_i)^T\big)\mathbf{C}_i^{-1} \\
&- \frac{1}{2}\mathbf{C}_i^{-1} + \frac{1}{2}\mathbf{C}_i^{-1}\big(\sum_k \beta_k\gamma_{i,k}(\mu_i-\mathbf{m}_k)(\mu_i-\mathbf{m}_k)^T\big)\mathbf{C}_i^{-1} \\
&- \frac{1}{2}\sum_k \gamma_{i,k}(\alpha_k+d+1)\mathbf{C}_i^{-1} + \frac{1}{2}\sum_k \gamma_{i,k}\mathbf{C}_i^{-1}\Lambda_k\mathbf{C}_i^{-1} = 0 \\
&\implies -\frac{N_i}{2}\mathbf{C}_i + \frac{1}{2}\sum_n (\mathbf{x}_{i,n}-\mu_i)(\mathbf{x}_{i,n}-\mu_i)^T - \frac{1}{2}\mathbf{C}_i \\
&+ \frac{1}{2}\sum_k \beta_k\gamma_{i,k}(\mu_i-\mathbf{m}_k)(\mu_i-\mathbf{m}_k)^T \\
&- \frac{1}{2}\sum_k \gamma_{i,k}(\alpha_k+d+1)\mathbf{C}_i + \frac{1}{2}\sum_k \gamma_{i,k}\Lambda_k = 0 \\
&\implies (N_i+1+\sum_k \gamma_{i,k}\alpha_k + d+1)\mathbf{C}_i = \sum_n (\mathbf{x}_{i,n}-\mu_i)(\mathbf{x}_{i,n}-\mu_i)^T \\
&+ \sum_k \beta_k\gamma_{i,k}(\mu_i-\mathbf{m}_k)(\mu_i-\mathbf{m}_k)^T + \sum_k \gamma_{i,k}\Lambda_k \\
&\implies \hat{\mathbf{C}}_i = \\
&\frac{\sum_{n=1}^{N_i}(\mathbf{x}_{i,n}-\mu_i)(\mathbf{x}_{i,n}-\mu_i)^T + \sum_{k=1}^{K}\beta_k\gamma_{i,k}(\mu_i-\mathbf{m}_k)(\mu_i-\mathbf{m}_k)^T + \sum_{k=1}^{K}\gamma_{i,k}\Lambda_k}{N_i + \sum_{k=1}^{K}\gamma_{i,k}\alpha_k + d + 2}
\end{aligned}
$$

Evaluating $\hat{\alpha}_k$

The below stated $f(\alpha_k)$ must be maximised w.r.t. α_k:

$$
\begin{aligned}
f(\alpha_k) = \frac{1}{2}\alpha_k \log|\Lambda_k|\sum_i \gamma_{ik} - \frac{d}{2}\log(2)\alpha_k\sum_i \gamma_{ik} - \\
\log\left(\Gamma_d\big(\frac{\alpha_k}{2}\big)\right)\sum_i \gamma_{ik} - \frac{1}{2}(\alpha_k+d+1)\sum_i \gamma_{ik}\log|C_i|
\end{aligned}
\tag{A.5}
$$

But the presence of $\log\left(\Gamma_d\big(\frac{\alpha_k}{2}\big)\right)\sum_i \gamma_{ik}$ term makes differentiation w.r.t. α_k complex. Therefore, a nonlinear optimisation must be used for evaluating α_k values at the M-step of every iteration. For the experiments discussed in this paper, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm was used to minimise $-f(\alpha_k)$, with limits set as $\alpha_k \in (d, d+20)$.

Evaluating $\hat{\pi}_k$

Evaluating $\hat{\pi}_k$ is a constrained optimisation problem, because π_k also have to satisfy an additional condition of $\sum_k \pi_k = 1$. Therefore, we need to maximise $[Q(\theta, \theta^t) + \eta(\sum_k \pi_k - 1)]$ w.r.t. π_k, where η is the Lagrange multiplier. From equation A.4, we have:

$$
\begin{aligned}
\frac{\partial[Q(\theta, \theta^t) + \eta(\sum_k \pi_k - 1)]}{\partial \pi_k} &\implies \frac{\sum_i \gamma_{i,k}}{\pi_k} + \eta = 0 \\
&\implies \pi_k = \frac{-\sum_i \gamma_{i,k}}{\eta}
\end{aligned}
$$

But since $\sum_k \pi_k = 1$; $\eta = \eta(\sum_k \pi_k) = -\sum_i \sum_k \gamma_{i,k}$ (from above) $= -I$ (by definition, because these are also the expectations of $\mathbf{z}_{i,k}$) where I are total assets in the fleet. Substituting value of η in above equation, we get:

$$
\hat{\pi}_k = \frac{\sum_{i=1}^{I} \gamma_{i,k}}{I}
$$

Appendix B

Proof for the Chi-squared Nature of the Squared Mahalanobis Distance

Proof for the standard chi-squared nature of the squared Mahalanobis distances (D^2_{md}) of points with respect to a d dimensional multivariate Gaussian is presented here. This proof is provided for the sake of completeness, where basic knowledge of linear algebra is assumed. The reader is advised to refer [175] for the complete derivation, and also the empirical proof.

For any given point X in space, its squared Mahalanobis distance (D^2_{md}) with respect to a multivariate Gaussian with mean μ and covariance Σ is evaluated as (assuming orthonormal eigenvectors):

$$D^2_{md} = (X - \mu)^T \Sigma^{-1} (X - \mu)$$

Upon performing he eigenvalue decomposition of Σ^{-1}, one obtains:

$$\Sigma^{-1} = U\Lambda^{-1}U^{-1} = U\Lambda U^T = \sum_{k=1}^{d} \lambda_k^{-1} u_k u_k^T$$

Where u_k is the k^{th} eigenvector of the corresponding eigenvalue λ_k.

Therefore,

$$
\begin{aligned}
D_{md} &= (X-\mu)^T \Sigma^{-1} (X-\mu) && \text{(B.1)} \\
&= (X-\mu)^T \Big(\sum_{k=1}^{d} \lambda_k^{-1} u_k u_k^T \Big) (X-\mu) && \text{(B.2)} \\
&= \sum_{k=1}^{d} \lambda_k^{-1} (X-\mu)^T u_k u_k^T (X-\mu) && \text{(B.3)} \\
&= \sum_{k=1}^{d} \lambda_k^{-1} \Big[\mu_k^T (X-\mu) \Big]^2 && \text{(B.4)} \\
&= \sum_{k=1}^{d} \Big[\lambda_k^{\frac{-1}{2}} \mu_k^T (X-\mu) \Big] && \text{(B.5)} \\
&= \sum_{k=1}^{d} Y_k^2 && \text{(B.6)}
\end{aligned}
$$

Where Y_k is a new random variable based on affine linear transformation of the random vector X.

We know that a random variable $Z = (X-\mu)$ can be expressed as $Z \sim \mathcal{N}(0, \Sigma)$. Similarly, the random variable Y_k introduced in (38) is of the form $Y_k = \lambda_k^{\frac{-1}{2}} \mu_k^T Z$. It can therefore be expressed as $Y_k \sim \mathcal{N}(0, \Sigma_k^2)$ where:

$$
\begin{aligned}
\Sigma_k^2 &= \lambda_k^{\frac{-1}{2}} u_k^T \Sigma \lambda_k^{\frac{-1}{2}} u_k \\
&= \lambda_k^{-1} u_k^T \Sigma \mu_k
\end{aligned}
$$

Upon substituting $\Sigma = \sum_{j=1}^{d} \lambda_j u_j u_j^T$,

$$
\begin{aligned}
\Sigma_k^2 &= \lambda_k^{-1} u_k^T \Sigma \mu_k \\
&= \lambda_k^{-1} u_k^T \Big(\sum_{j=1}^{d} \lambda_j u_j u_j^T \Big) \mu_k \\
&= \sum_{j=1}^{d} \lambda_k^{-1} u_k^T \lambda_j u_j u_j^T u_k \\
&= \sum_{j=1}^{d} \lambda_k^{-1} \lambda_j u_k^T u_j u_j^T u_k
\end{aligned}
$$

Since all eigenvectors u_i are pairwise orthonormal, the dotted products $u_k^T u_j$ and $u_j^T u_k$ will be zero for $j \neq k$. Only for the case $j = k$ we get:

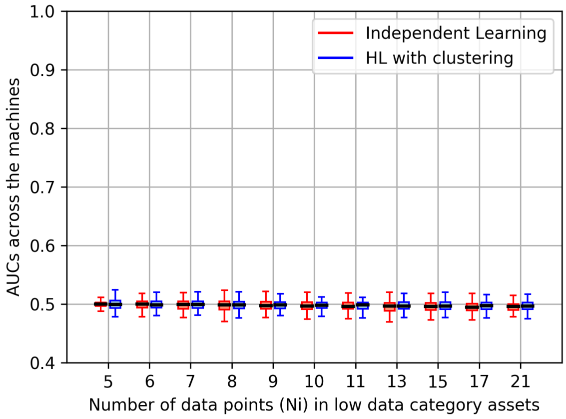
$$\begin{aligned}\Sigma_k^2 &= \lambda_k^{-1} \lambda_k u_k^T u_k u_k^T u_k \\ &= \lambda_k^{-1} \lambda_k ||u_k||^2 ||u_k||^2 \\ &= 1\end{aligned}$$

The last step follows because the norm $||u_k||$ of an orthonormal eigenvector is equal to 1. The squared D_{md} can thus be expressed as $D_{md}^2 = \sum_{k=1}^{d} Y_k^2$ where $Y_k \sim \mathcal{N}(0,1)$. This is also the exact definition of a standard chi-squared distribution with d degrees of freedom, i.e. the sum of the squared of d random variables which are standard normally distributed. Therefore, the squared D_{md} is chi-squared with d degrees of freedom and can therefore be used to obtain a critical value for anomaly detection.

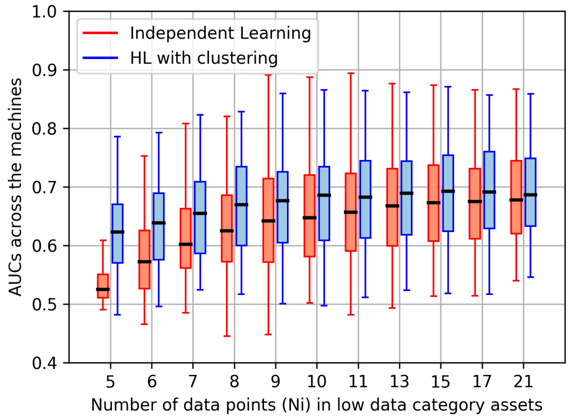
Appendix C

Extended Results From the Experiments in Chapter 3

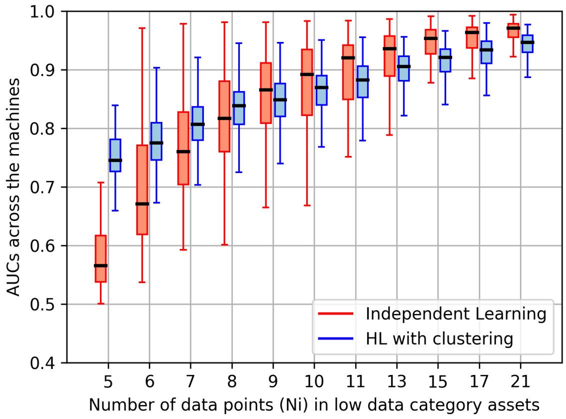
C.1 Results demonstrating the benefit of hierarchical modelling for the low data category assets.



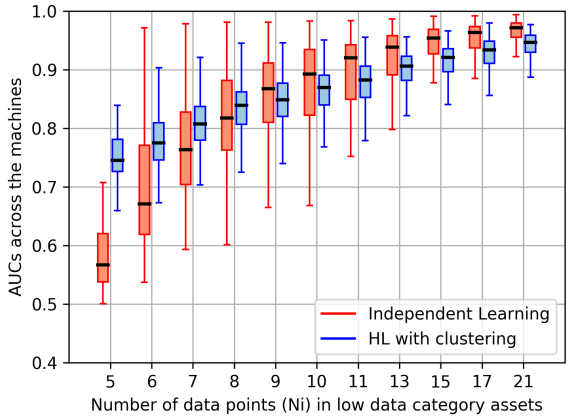
(a) $l = 0; L = 1$



(b) $l = 5; L = 1$

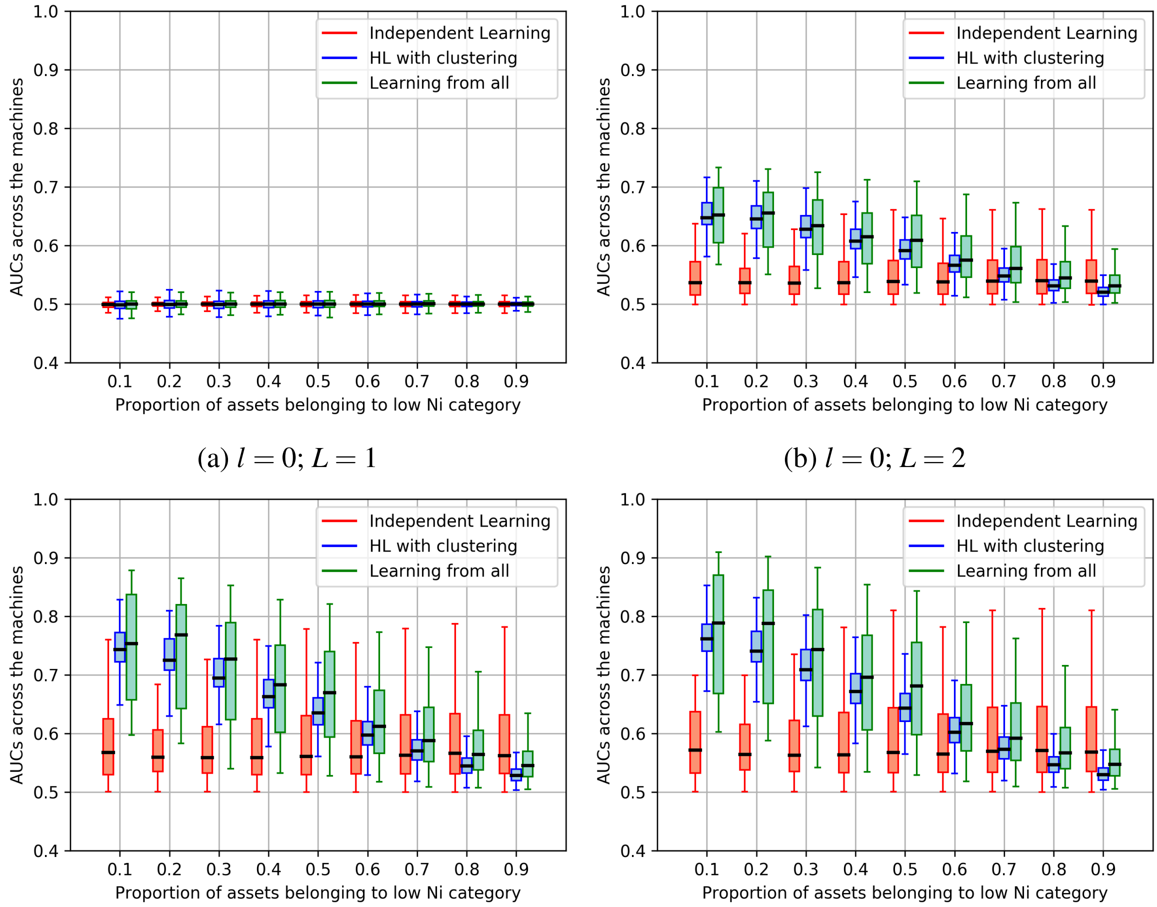


(c) $l = 20; L = 1$



(d) $l = 100; L = 1$

Fig. C.1 Box plots presenting the effect of gradually increasing data contained by the low data category assets. The captions denote the corresponding deviations in the testing dataset



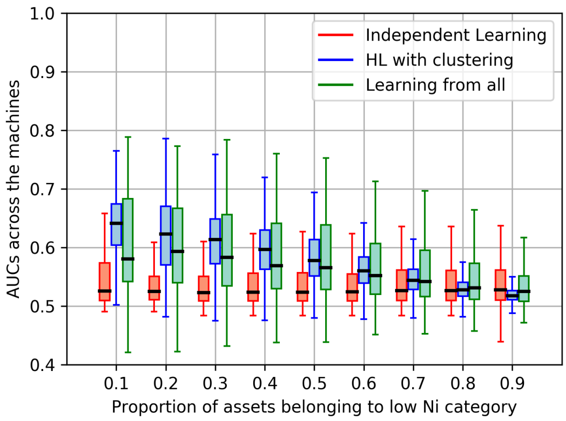
(a) $l = 0; L = 1$

(b) $l = 0; L = 2$

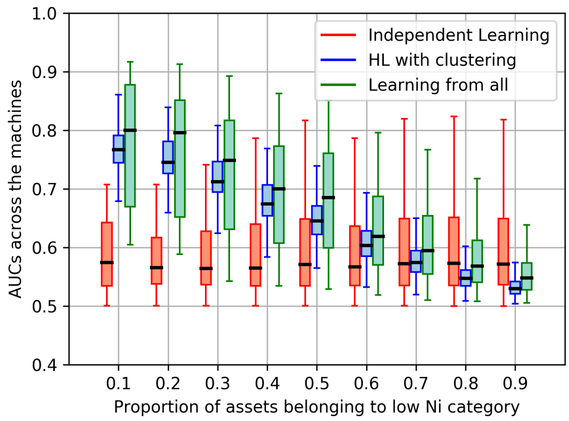
(c) $l = 0; L = 5$

(d) $l = 0; L = 10$

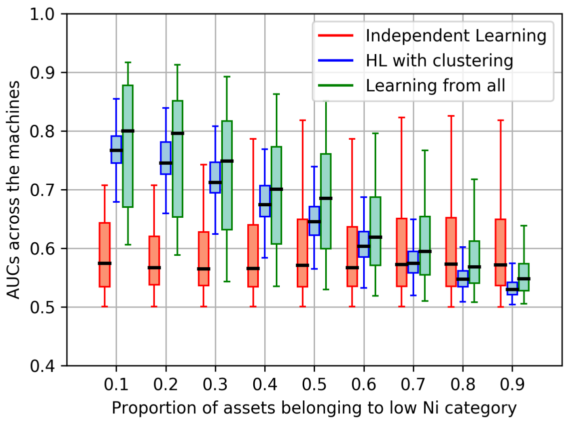
Fig. C.2 Box plots presenting AUCs recorded across the assets belonging to the low data category. The corresponding testing dataset deviations are denoted in the captions



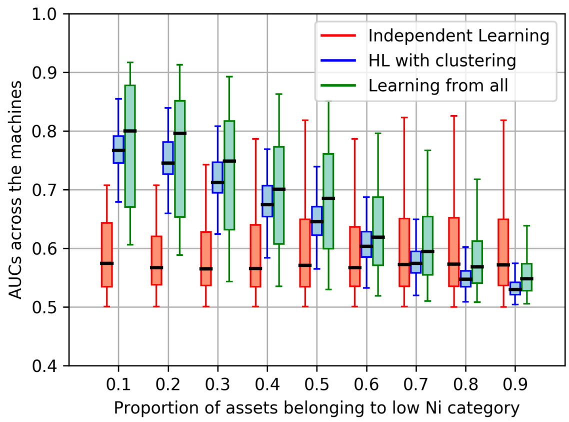
(a) $l = 5$; $L = 1$



(b) $l = 20$; $L = 1$



(c) $l = 50$; $L = 1$

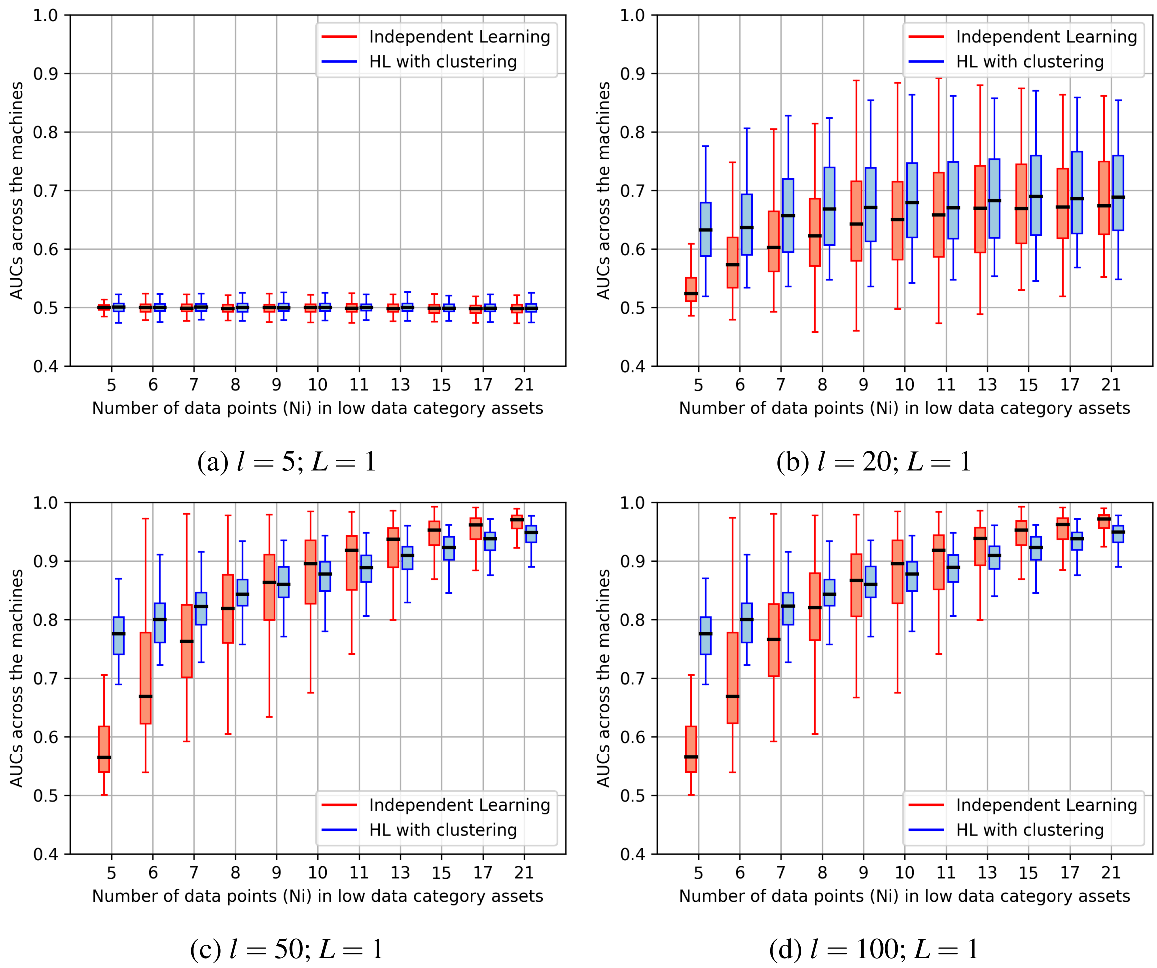


(d) $l = 100$; $L = 1$

Fig. C.3 Box plots presenting AUCs recorded across the assets belonging to the low data category. The corresponding testing dataset deviations are denoted in the captions

C.2 Results from the experiment conducted for a shorter range of asset means.

Figure C.4 shows the comparison of performances of the hierarchical model and independent learning for the clusters with a narrow range of means representing the asset model types. The asset clusters comprised of means ranging within $(-5, 5)$ for one model type and $(295, 305)$ for the other. The covariance matrices used to generate data were the same as the ones shown in (3.25 and 3.25). A slight improvement in performance of the hierarchical model can be observed, due to the fact that the assets in a cluster here are more similar to one another. Figure C.4 is evaluated in the same manner as Figure C.1 but for the training and testing datasets corresponding to a narrower range of means.



(a) $l = 5$; $L = 1$ (b) $l = 20$; $L = 1$

(c) $l = 50$; $L = 1$ (d) $l = 100$; $L = 1$

Fig. C.4 Box plots presenting AUCs recorded across the assets belonging to the low data category, but for a narrower range of means