Data-Centric Engineering (2020), **xx**:xx 1–34
doi:10.1017/xxxx

RESEARCH ARTICLE

Anomaly Detection in a Fleet of Industrial Assets with Hierarchical Statistical Modelling

Maharshi Dhada[1*], Mark Girolami[2,3] and Ajith Kumar Parlikad[1]

[1]Institute for Manufacturing, Department of Engineering, University of Cambridge, 17 Charles Babbage Rd, Cambridge, CB3 0FS, U.K. [2]Department of Engineering, University of Cambridge, Cambridge, CB3 0FA, U.K. [3]The Alan Turing Institute, London, NW1 2DB, U.K.
*Corresponding author. E-mail: mhd37@cam.ac.uk

(Received xx xxx xxxx)

Keywords: Hierarchical Modelling; Anomaly Detection; Reliability Engineering; Health Management; Machine Learning

Abstract
Anomaly detection in condition data is critical for reliable industrial asset operations. But statistical classifiers require certain amount of normal operations data before acceptable accuracy can be achieved. The necessary training data however is often not available in the early period of asset operations. This problem is addressed here by using a hierarchical model for asset fleet that systematically identifies similar assets, and formulates higher level distributions of the asset level parameters. Hierarchical models enable the individuals from a population, comprising of statistically coherent sub-populations, to collaboratively learn from one another. The higher level distributions in this paper represent the general behaviour of similar assets, and the individual asset behaviours are described by the parameters sampled from higher level distributions. 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.

Impact Statement
It is shown in this paper that industrial assets with low amount of data can significantly improve the performances of their anomaly detection classifiers by collaborating with similar assets containing more data. The authors enable this collaborative learning via a hierarchical model of the asset fleet, that defines higher level distributions representing the general behaviour of asset clusters and individual asset level parameters sampled from the higher level distributions.

1. Introduction
Modern industrial assets are embedded with a plethora of sensors that monitor asset operations in real time. Availability of asset condition time series combined with readily available computing power and communication technologies has extensively automated industrial operations in the recent decade (Gilchrist and Gilchrist, 2016; Xu et al., 2014).

Asset health management in particular has moved from physics based formulations to Machine Learning (ML) techniques. As a part of asset health management, detecting anomalies in an asset's condition data is important for accurate prognosis. An ideal anomaly detection algorithm instantaneously identifies deviations in real time, and activates the prognosis algorithm to plan timely maintenance. Accurate anomaly detection also enables efficient extraction of the failure trajectories from historical condition data. Failure trajectories are the time series ranging from the asset's deviation from normal behaviour till its failure. Since historical failure trajectories constitute the training dataset for prognosis, learning capabilities of the prognosis models primarily depend on accurate anomaly detection. An inefficient anomaly detection algorithm instead could let a failure go undetected, or flag many anomalies that turn out to be benign and not require any intervention (Kang, 2018).

Most industries today rely on rule based systems for anomaly detection. These comprise of preset warnings and trip limits on the sensor measurements (Zaidan et al., 2015; Saxena et al., 2008). Force tripping an asset often results in production losses, which could have been avoided if a planned maintenance was carried out in good time. Moreover, the warning-trip systems are inherently non-responsive. 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 start exceeding the preset limits, which might already be too late and the opportune time be lost.

In scenarios where the domain knowledge about the underlying distribution is available beforehand, statistical classifiers provide a mathematically justifiable solution for anomaly detection. Statistical classifiers posit that the condition monitoring data generated during normal asset operations can be described using a family of underlying distributions. Assuming that an asset commences operating in normal condition, the underlying probability 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 underlying estimated density function. Statistical tests are used to evaluate if a newly recorded data point is significantly different to be deemed anomalous or not (Kang, 2018; Rajabzadeh et al., 2016).

Statistical classifiers are amongst the recommended anomaly detection techniques in the recent literature on asset health management (Kang, 2018). 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. For most preliminary algorithms deployment and simulations, the combined random effect of error and fluctuations in the sensor measurements has been treated as multivariate Gaussian (Borguet and Léonard, 2009; Kobayashi and Simon, 2005; Saxena et al., 2008).

But independent modelling of assets is accompanied with challenges, primarily that of 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, owing to the statistically heterogeneous nature of asset operations, collective modelling of the fleet wide data is challenging (Salvador Palau et al., 2019). These characteristics impede the application of statistical classifiers for detecting anomalies in the early periods of asset operations when sufficient training data is not available. Therefore, a systematic method for modelling the underlying clusters of similar assets, and enabling their comprising assets to collaboratively learn from one another is much needed.

This paper addresses the above problem by using a hierarchical model for the asset fleet that systematically identifies similar assets, and formulates higher level distributions of the asset level parameters. Hierarchical models enable the individuals from a population, comprising of statistically coherent sub-populations, to collaboratively learn from one another. (Gelman et al., 2013; Eckert et al., 2007; Hensman et al., 2013; Teacy et al., 2012). The higher level distributions in this paper represent the general behaviour of similar assets, and the individual asset behaviours are described by the parameters sampled from higher level distributions. Comprehensive information about hierarchical modelling can be found in (Gelman et al., 2013; Gelman and Hill, 2006).

The continuing paper is structured as: Section 2 discusses the prevalent hierarchical modelling and collaborative anomaly detection techniques specifically in the industrial health management literature. Following this, Section 3 describes hierarchical modelling of an asset fleet, including the mathematical description for extending an asset's independent model to a hierarchical fleet-wide model containing clusters of similar assets. An example implementation of the hierarchical model for a simulated fleet of assets is shown in Section 4. The same section also compares the performance of the hierarchical model with the case where the asset parameters were independently estimated. The results from the experiments are discussed in Section 5. Lastly, Sections 6 and 7 summarise the key conclusion and highlight the future research directions respectively.

2. Literature Review

This section discusses the prevalent applications of hierarchical modelling and automated anomaly detection in the context of industrial assets' health management.

2.1. Hierarchical Modelling of the Industrial Assets

Applied mathematicians have stressed on understanding the heterogeneous nature of the industrial assets since as long as 1967. Lindley et al. (1967) 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. Ascher (1983) 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. Ascher (1983) showed that using the trend test proposed by Lindley et al. (1967) followed by a non homogeneous poisson processes model, independent industrial assets could be described significantly more accurately.

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 (Arjas and Bhattacharjee, 2004). For the modern industrial automation almost entirely relying on data-driven ML algorithms, such oblivion to the statistically heterogeneous nature of industrial data poses ever greater risk. Industrial automation, according to the notion of Industry 4.0, aims at end-to-end hands off collaborative control made possible by a series of decision making algorithms (Gilchrist and Gilchrist, 2016). For example, a maintenance planning procedure broadly comprises of anomaly detection, followed by failure prediction, followed by maintenance planning, and finally followed by resource allocation. In such a serial dependency, inefficiencies or inaccuracies of an algorithm governing any of these steps can easily perpetuate through the control pipeline and deteriorate the decision making of the algorithms in the following steps.

Industrial asset fleets are in fact a collection of not identical, but similar individuals. For example, a collection of automobiles could be manufactured differently but they all share similarities in their basic design (Chen and Singpurwalla, 1996). This characteristic make hierarchical models a suitable solution for statistical analyses of the asset fleets. While modelling the asset fleets, collective behaviours of clusters of similar assets are described using higher level distributions, from which are sampled the parameters describing individual asset operations. For the asset health management applications, researchers have proposed using hierarchical modelling to account for system heterogeneity. While most of the applications focus on describing times between failures, there are also some instances in recent literation where the condition data-driven real time prognosis is enhanced using hierarchical modelling.

One of the earliest applications use hierarchical bayesian estimation of Bernoulli model parameters for reliability estimation of emergency diesel generators in separate nuclear power plants (Chen and Singpurwalla, 1996). They showed that hierarchical Bernoulli model was a better technique for simultaneously 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. Most other applications in the traditional survival analysis target modelling the times between failures, similar to the illustration described in (Ascher, 1983). For example, Arjas and Bhattacharjee (2004) 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. An interesting application can also be found in (Johnson et al., 2005) where hierarchical modelling was used for reliability estimation of new space crafts, which had experienced none to few failures. Similar other applications include (Economou et al., 2007; Dedecius and Ettler, 2014; Yuan and Ji, 2015), all commonly modelling the times between failures for various equipment.

Of the more recent but fewer condition data-driven prognosis applications, Zaidan et al. (2015) 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. Kao and Chen (2012) 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.

2.2. Anomaly Detection for Industrial Assets

The traditional applications of anomaly detection mostly target system diagnostics, involving fault identification and classification. However, with condition data readily available, online anomaly detection techniques are recently gaining popularity.

Anomaly detection in industrial asset operations is challenging. This is because the assets operate over a wide range of environments, in various operating regimes, and can fail in multiple modes (Khan and Madden, 2010; Michau and Fink, 2019). Every asset has its own unique behaviour and failure tendency, and therefore requires an anomaly detector particularly suited for its operations. Moreover, the assets do not fail frequently, making the classifier's training data highly imbalanced towards "normal operation" class. Researchers therefore often treat anomaly detection in asset operations as a one-class time series classification problem (Kang, 2018).

This paper focuses only on the statistical classifiers, which are introduced in Section 1, due to their straightforward implementation compared to more sophisticated algorithms like deep learning. Such statistical classifiers have been proposed by several researchers for anomaly detection in gas turbine combustors, cooling fans, and general performance monitoring (Kang, 2018; Borguet and Léonard, 2009; Yan, 2016; Jin et al., 2012).

Interestingly, the literature presents examples where different degrees and forms of collaboration amongst the assets have shown to improve the performances of anomaly detectors. In the simplest form of collaboration, 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 IID dataset. This type of collaboration can be found in (Zio and Di Maio, 2010; González-Prida et al., 2016; Lapira and Lee, 2012), where in every case the operators use a relevant parameter for clustering the corresponding assets. Some researchers have also clustered the entire time series of condition monitoring data based on their Euclidean distances like in the case of (Liu, 2018; Leone et al., 2016; Al-Dahidi et al., 2018). In a comparatively more complex collaborative approach, Michau et al. (2018) 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 (Zio and Di Maio, 2010; González-Prida et al., 2016; Lapira and Lee, 2012), dimensional complexity while evaluating the Euclidean distances in (Liu, 2018; Leone et al., 2016; Al-Dahidi et al., 2018), and the necessary training data for each unit required to train the neural networks in the case of (Michau et al., 2018).

Amongst examples of collaborative anomaly detection solutions, the closest one to the problem discussed in this paper can be found in Michau and Fink (2019). Michau and Fink (2019) stress the necessity of one class-classification for industrial systems owing to a wide range of possible operating regimes and rarity of failures. Michau and Fink (2019) 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, Michau and Fink (2019) focus on feature alignment and the current paper focuses on modelling an overall fleet behaviour and modifying it to suit individual assets. As such, the solution proposed in this paper differs from the one presented in Michau and Fink (2019) in three aspects. First, the proposed hierarchical model is capable of identifying the asset clusters in the fleet, in contrast to Michau and Fink (2019) where it is assumed that all assets within the fleet are similar or known beforehand. Second, the operating regime targeted in this paper is that of earlier operations compared to Michau and Fink (2019), where the assets they describe as new have 17,000 data points for 24-dimensional data. Lastly, hierarchical modelling presented here is a distributed learning technique, and more importantly a technique that enables the assets to learn from each others models rather than their data.

In summary, 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. The authors believe that hierarchical modelling of the asset fleet addresses this challenge by enabling the assets with insufficient data to collaborate with other similar assets containing more data. The literature also shows that hierarchical modelling is a reliable technique to model heterogeneity in an asset fleet but, to the best of the authors' knowledge, it has not yet been implemented for data-driven anomaly detection in industrial assets.

3. Mathematical Description

3.1. Independent asset models

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

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

Maximum likelihood estimation can be used to evaluate $\hat{\boldsymbol{\mu}}_i$ and $\hat{\mathbf{C}}_i$ values for $\mathbf{X}_i$. A graphical representation of an isolated independent asset model is shown in Figure 1. The following section describes extending the independent asset model to a hierarchical model.

3.2. Hierarchical modelling

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 (Jin et al., 2015; Leone et al., 2017). It gives rise to the presence of statistically homogenous *asset clusters* within the fleet. The challenges related to distribution instabilities mentioned in Section 1 can be alleviated if the

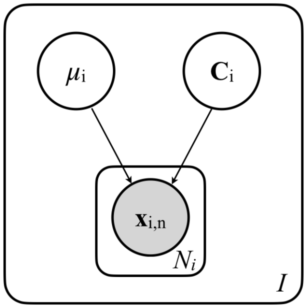


Figure 1. Graphical representation of modelling an asset's data as multivariate Gaussian.

individuals comprising such a cluster are jointly modelled with a common underlying distribution of their individual distribution parameters.

Hierarchical model of the asset fleet 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 $(\boldsymbol{\mu}_i, \mathbf{C}_i)$ describing i^{th} asset are believed to be sampled from higher distributions as $\boldsymbol{\mu}_i \sim \mathcal{N}(\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)$ and $\mathbf{C}_i \sim \mathcal{IW}(\boldsymbol{\Lambda}_k, \alpha_k)$ where $k = 1, 2, ..., K$ represents the cluster index and $(\mathbf{m}_k \in \mathbb{R}^d, \beta_k \in \mathbb{R}, \boldsymbol{\Lambda}_k \in \mathbb{R}^{d\times d}, \alpha_k \in \mathbb{R})$ are the parameters of cluster level distributions.

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

$$p(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k) = \mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k) = \frac{|\boldsymbol{\Lambda}_k|^{\alpha_k/2}}{2^{\alpha_k d/2}\Gamma_d(\frac{\alpha_k}{2})}|\mathbf{C}_i|^{-(\alpha_k+d+1)/2} \exp\Big(-\frac{1}{2}Tr(\boldsymbol{\Lambda}_i\mathbf{C}_i^{-1})\Big) \tag{3}$$

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 $(\boldsymbol{\mu}_i, \mathbf{C}_i)$ are sampled. The probability density function for a given $(\boldsymbol{\mu}_i, \mathbf{C}_i)$ pair conditional on higher level parameters therefore can therefore be written as:

$$p(\boldsymbol{\mu}_i, \mathbf{C}_i|\mathbf{m}_k, \beta_k, \boldsymbol{\Lambda}_k, \alpha_k) = \sum_{k=1}^{K}\Big[\pi_k\mathcal{N}(\boldsymbol{\mu}_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k)\Big] \tag{4}$$

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 $(\boldsymbol{\mu}_i, \mathbf{C}_i)$ pair.

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

$$p(\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, ..., \mathbf{x}_{1,N_i}) = \prod_{n=1}^{N_i} \Big[\mathcal{N}(\boldsymbol{\mu}_i, \mathbf{C}_i) \sum_{k=1}^{K} \big[\pi_k \mathcal{N}(\boldsymbol{\mu}_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k)\big]\Big] \tag{5}$$

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

$$p(\mathbf{X}) = \prod_{i=1}^{I} \Bigg[\prod_{n=1}^{N_i} \Big[\mathcal{N}(\boldsymbol{\mu}_i, \mathbf{C}_i) \sum_{k=1}^{K} \big[\pi_k \mathcal{N}(\boldsymbol{\mu}_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k)\big]\Big]\Bigg] \tag{6}$$

For a given set of $(\boldsymbol{\mu}_i, \mathbf{C}_i, \mathbf{m}_k, \alpha_k,)$, the above function is also the likelihood of the data. Obtaining estimates of $(\boldsymbol{\mu}_i, \mathbf{C}_i, \mathbf{m}_k, \alpha_k,)$ 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:

$$\log(p(\mathbf{X})) = \sum_{i=1}^{I}\sum_{n=1}^{N_i} \log(\mathcal{N}(\boldsymbol{\mu}_i, \mathbf{C}_i)) + \sum_{i=1}^{I} \log\Big(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\mu}_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k)\Big) \tag{7}$$

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.

3.2.1. Model Parameters Estimation

Maximising the log-likelihood in (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:

$$p(\mathbf{z}_{i,k}|\boldsymbol{\theta}) = \pi_k \tag{8}$$

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

$$p(\mathbf{z}_i|\boldsymbol{\theta}) = \prod_{k=1}^{K} [\pi_k]^{\mathbf{z}_{i,k}} \tag{9}$$

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

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

$$p(\boldsymbol{\mu}_i, \mathbf{C}_i|\mathbf{z}_{i,k} = 1, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k) \tag{10}$$

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

$$p(\boldsymbol{\mu}_i, \mathbf{C}_i|\mathbf{z}_i = 1, \boldsymbol{\theta}) = \prod_{k=1}^{K} \big[\mathcal{N}(\boldsymbol{\mu}_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k)\big]^{\mathbf{z}_{i,k}} \tag{11}$$

Probability of $(\boldsymbol{\mu}_i, \mathbf{C}_i, \mathbf{z}_i)$ can therefore be evaluated simply by multiplying (9) and (11) as:

$$p(\boldsymbol{\mu}_i, \mathbf{C}_i, \mathbf{z}_i|\boldsymbol{\theta}) = \prod_{k=1}^{K} \big[\pi_k \mathcal{N}(\boldsymbol{\mu}_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k)\big]^{\mathbf{z}_{i,k}} \tag{12}$$

Continuing similar to (5) and (6), the complete data probability for a given set of parameters $\boldsymbol{\theta}$ is given by:

$$p(\mathbf{X}, \mathbf{z}|\boldsymbol{\theta}) = \prod_{i=1}^{I} \left[\prod_{n=1}^{N_i} \left[\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_i, \mathbf{C}_i) \right] \prod_{k=1}^{K} \left[\pi_k \mathcal{N}(\boldsymbol{\mu}_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i) \mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k) \right]^{\mathbf{z}_{i,k}} \right] \tag{13}$$

The graphical representation shown in Figure 2 describes the hierarchical modelling for whole fleet data, including the hidden cluster indicator $\mathbf{z}$.

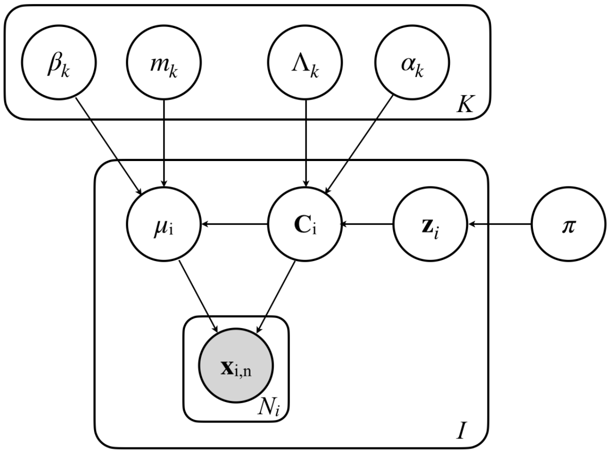


Figure 2. Graphical representation of hierarchically modelled fleet data. Individual asset data are modelled as multivariate Gaussians, whose mean and covariance parameters are sampled from higher level Normal-Inverse Wishart distributions respectively.

The complete data log-likelihood for a given set of parameters $\boldsymbol{\theta}$ thus equates to:

$$\log(p(\mathbf{X}, \mathbf{z}|\boldsymbol{\theta})) = \sum_{i=1}^{I} \sum_{n=1}^{N_i} \log(\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_i, \mathbf{C}_i)) + \sum_{i=1}^{I} \sum_{k=1}^{K} \mathbf{z}_{i,k} \log \left(\pi_k \mathcal{N}(\boldsymbol{\mu}_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i) \mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k) \right) \tag{14}$$

To maximise the complete data log-likelihood function in (14), (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, $\boldsymbol{\theta}$ are the model parameters and the parameters corresponding to t^{th} iteration are written as $\boldsymbol{\theta}^t$.

In the E-step, a function $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t)$ 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 $\boldsymbol{\theta}^t$ parameter values. Concretely,

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) = E_{z|X,\boldsymbol{\theta}^{t-1}}\{\log(l(\mathbf{X}, \mathbf{z}|\boldsymbol{\theta}))\} \tag{15}$$

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

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

$$\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) \tag{16}$$

Estimated values of model parameters at M-step of every EM iteration are presented in (17) to (22), where the "$\boldsymbol{\gamma}_{i,k}$" 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.

$$\frac{1}{\hat{\beta}_k} = \frac{\sum_{i=1}^{I} \boldsymbol{\gamma}_{i,k} (\boldsymbol{\mu}_i - \mathbf{m}_k)^T \mathbf{C}_i^{-1} (\boldsymbol{\mu}_i - \mathbf{m}_k)}{d \sum_{i=1}^{I} \boldsymbol{\gamma}_{i,k}} \tag{17}$$

$$\hat{\mathbf{m}}_k = \left[\sum_{i=1}^{I} \boldsymbol{\gamma}_{i,k} \mathbf{C}_i^{-1} \right]^{-1} \left[\sum_{i=1}^{I} \boldsymbol{\gamma}_{i,k} \mathbf{C}_i^{-1} \boldsymbol{\mu}_i \right] \tag{18}$$

$$\hat{\boldsymbol{\Lambda}}_k = \left[\alpha_k \sum_{i=1}^{I} \boldsymbol{\gamma}_{i,k} \right] \left[\sum_{i=1}^{I} \boldsymbol{\gamma}_{i,k} \mathbf{C}_i^{-1} \right]^{-1} \tag{19}$$

$$\hat{\boldsymbol{\pi}}_k = \frac{\sum_{i=1}^{I} \boldsymbol{\gamma}_{i,k}}{I} \tag{20}$$

$$\hat{\boldsymbol{\mu}}_i = \frac{1}{N_i + \sum_{k=1}^{K} \beta_k \boldsymbol{\gamma}_{i,k}} \left[\sum_{n=1}^{N_i} \mathbf{x}_{i,n} + \sum_{k=1}^{K} \beta_k \boldsymbol{\gamma}_{i,k} \mathbf{m}_k \right] \tag{21}$$

$$\hat{\mathbf{C}}_i = \frac{\sum_{n=1}^{N_i} (\mathbf{x}_{i,n} - \boldsymbol{\mu}_i)(\mathbf{x}_{i,n} - \boldsymbol{\mu}_i)^T + \sum_{k=1}^{K} \beta_k \boldsymbol{\gamma}_{i,k} (\boldsymbol{\mu}_i - \mathbf{m}_k)(\boldsymbol{\mu}_i - \mathbf{m}_k)^T + \sum_{k=1}^{K} \boldsymbol{\gamma}_{i,k} \boldsymbol{\Lambda}_k}{N_i + \sum_{k=1}^{K} \boldsymbol{\gamma}_{i,k} \alpha_k + d + 2} \tag{22}$$

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 (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. $(\boldsymbol{\mu}_i, \mathbf{C}_i)\ \forall\ i \in \{I\}$) 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. $(\mathbf{m}_k, \boldsymbol{\Lambda}_k)\ \forall\ k \in \{K\}$ initialised randomly around the observed data values, but ensuring that the initial $\boldsymbol{\Lambda}_k$ were positive definite matrices. The steps followed for hierarchical model parameters estimation, including the initialisation in the experiments described here and EM iterations, are described in Algorithm 1. In Algorithm 1, $E(x_{i,n})$ in line 4 represents the expectation of $x_{i,n}$ vector, $rand(d)$ and $rand(d,d)$ functions in line 9 generate random real numbered matrices of (d) and $(d \times d)$ dimensions respectively, and $p(clust_i = k)$ in line 16 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 $\boldsymbol{\gamma}_{i,k}$ which are evaluated at the corresponding E-step.

Algorithm 1: Pseudo-code describing the steps to estimate the hierarchical model parameters for an asset fleet comprising K clusters and generating d dimensional condition data

Result: Estimated hierarchical model parameters

Initialise the parameters:

for *each asset i* **do**

  $\boldsymbol{\mu}_i \leftarrow \frac{\sum_{n=1}^{N_i} \boldsymbol{x}_{i,n}}{N_i}$;

  $\boldsymbol{C}_i^{(n,m)} \leftarrow E\big((\boldsymbol{x}_{i,n} - E(\boldsymbol{x}_{i,n}))(\boldsymbol{x}_{i,m} - E(\boldsymbol{x}_{i,m})\big)$;

end

for *each cluster k* **do**

  $\beta_k \leftarrow 0.001$;

  $\alpha_k \leftarrow d$;

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

end

The EM iterations:

while *Iter < 20* **do**

  The E-step:

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

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

  end

  The M-step:

  for *each asset i* **do**

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

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

  end

  for *each cluster k* **do**

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

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

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

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

    $\alpha_k \leftarrow \mathrm{BFGS}_{max}\bigg(\frac{1}{2}\alpha_k\log|\boldsymbol{\Lambda}_k|\sum_i\boldsymbol{\gamma}_{ik} - \frac{d}{2}\log(2)\alpha_k\sum_i\boldsymbol{\gamma}_{ik} - \log\Big(\Gamma_d\big(\frac{\alpha_k}{2}\big)\Big)\sum_i\boldsymbol{\gamma}_{ik} - \frac{1}{2}(\alpha_k+d+1)\sum_i\boldsymbol{\gamma}_{ik}\log|\boldsymbol{C}_i|\bigg)$;

  end

  $Iter \leftarrow Iter + 1$;

end

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

4. 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 modelling of the assets.

Independent modelling does not consider the presence of similar assets in the fleet. Therefore, the $(\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{C}}_i)$ 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 (23) and (24).

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum_{n=1}^{N_i} \boldsymbol{x}_{i,n}}{N_i} \tag{23}$$

$$\hat{\boldsymbol{C}}_i^{(n,m)} = E\big((\boldsymbol{x}_{i,n} - E(\boldsymbol{x}_{i,n}))(\boldsymbol{x}_{i,m} - E(\boldsymbol{x}_{i,m})\big) \tag{24}$$

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

Experimental cases, and the performance metric used for evaluating and comparing both modelling approaches are described in the following subsections. Section 4.1 explains the synthetic dataset used for the experiments, Section 4.2 briefly describes the Receiver Operating Characteristic curves which were used as an evaluation metric, and finally Sections 4.3 and 4.4 presents the experimental results to compare the performances of hierarchical and independent modelling techniques.

4.1. 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.

4.1.1. 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 as explained in Section 1. 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.

4.1.2. Testing dataset

The testing dataset for any given simulated asset described in Section 4.1.1 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. 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 for easier representation.

4.1.3. 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 25. The ranges for means and the two covariance matrices were arbitrarily chosen.

$$C^1 = \begin{bmatrix} 16.68 & 5.43 & 3.28 & -2.31 & 1.76 \\ 5.43 & 22.05 & -3.74 & -1.11 & -1.14 \\ 3.28 & -3.74 & 18.72 & 3.91 & -3.19 \\ -2.31 & -1.11 & 3.91 & 20.87 & 4.00 \\ 1.76 & -1.14 & -3.19 & 4.00 & 23.12 \end{bmatrix} \text{ and } C^2 = \begin{bmatrix} 55.59 & 3.39 & 3.24 & -2.00 & -3.95 \\ 3.39 & 55.75 & 1.22 & -24.02 & -3.76 \\ 3.24 & 1.22 & 55.83 & 15.29 & 1.78 \\ -2.00 & -24.02 & 15.29 & 63.69 & 11.21 \\ -3.95 & -3.76 & 1.78 & 11.21 & 23.12 \end{bmatrix} \tag{25}$$

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 are summarised in Table 1.

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

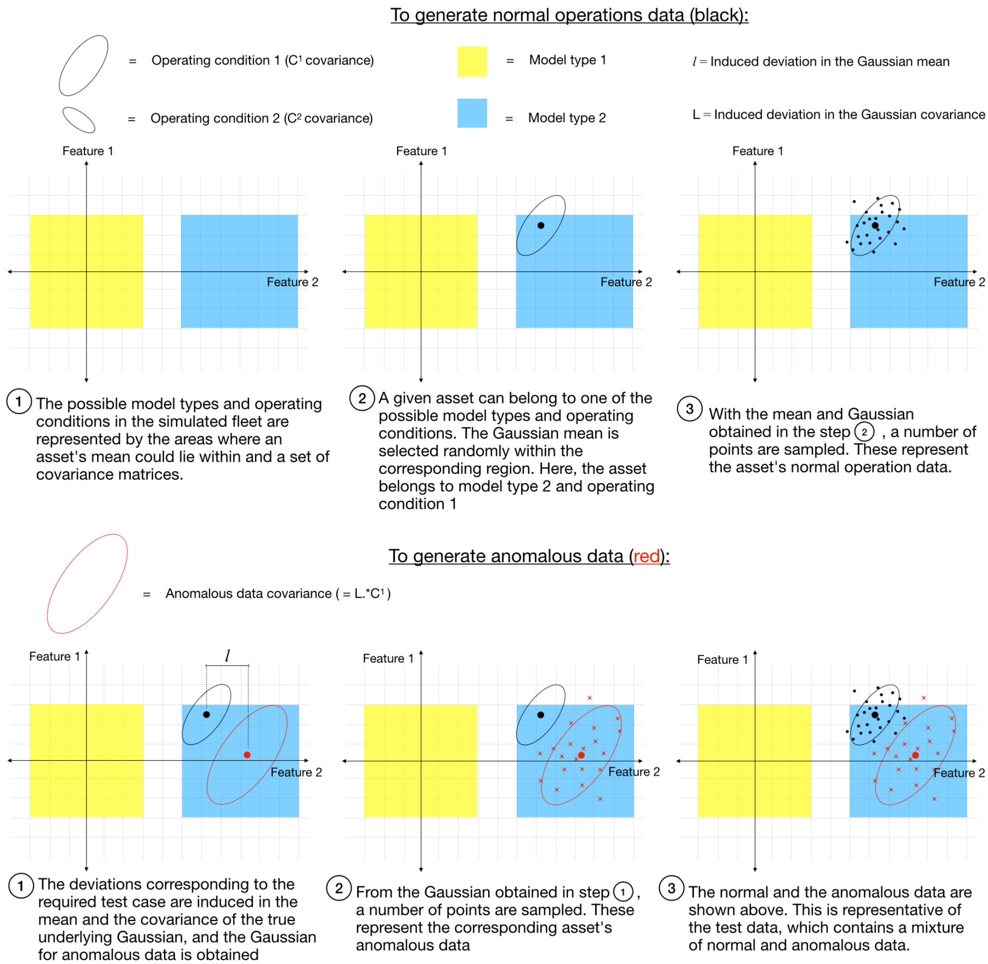


Figure 3. A schematic representation describing how the normal and anomalous data were generated for the experiments. The procedure is shown here for a two dimensional dataset as an example..

Table 1. The values of various parameters introduced in Section 3.

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)

$(10.05, -15.95, 4.94, -4.24, 0.68)$. Next, with this mean and C^1 from (25) 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 dataset can be found at: https://github.com/Dhada27/Hierarchical-Modelling-Asset-Fleets

Table 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.

4.2. 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 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.

[1]Proof shown in Appendix B

Table 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 $\hat{\boldsymbol{\mu}}$ and $\hat{\mathbf{C}}$ is obtained as:

$$D^2_{md} = (\mathbf{X} - \hat{\boldsymbol{\mu}})^T \hat{\mathbf{C}}^{-1} (\mathbf{X} - \hat{\boldsymbol{\mu}}) \tag{26}$$

Areas under the Receiver operator characteristic (ROC) curves were used as a measure for evaluating the performance of hierarchical modelling, and also for comparing with conventional independent modelling technique. This is a widely used evaluation metric for classification tasks and is 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 the “negative" class and those sampled from the anomalous distribution were labelled as the “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:

$$TPR = \frac{N_{TP}}{N_P} \qquad FPR = \frac{N_{FP}}{N_N} \tag{27}$$

The Area Under the ROC Curve (AUC) was used as an indicator of the model’s performance for a given asset. From (27), 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 4. This ROC was evaluated for the parameters estimated based on hierarchical modelling.

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

4.3. Experimental Design

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

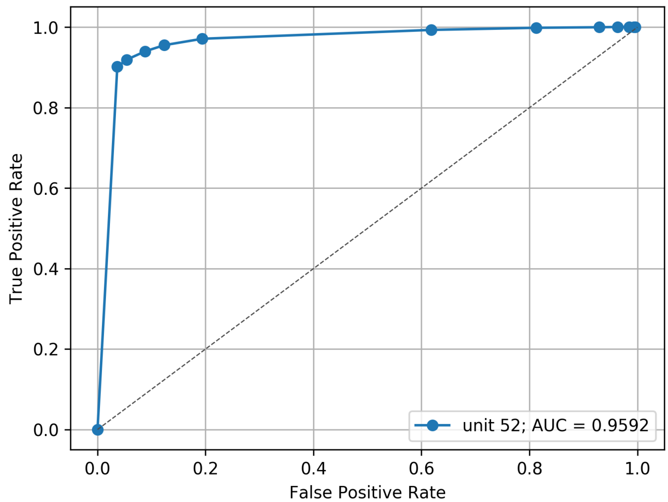
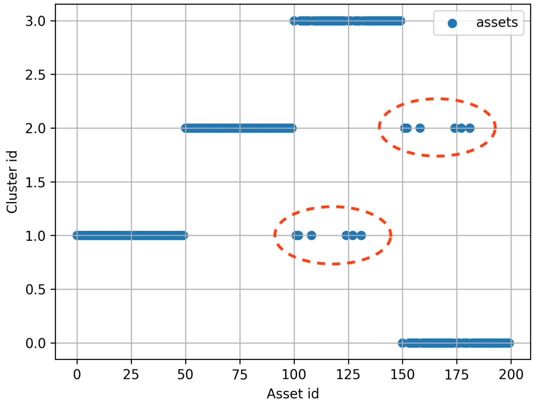


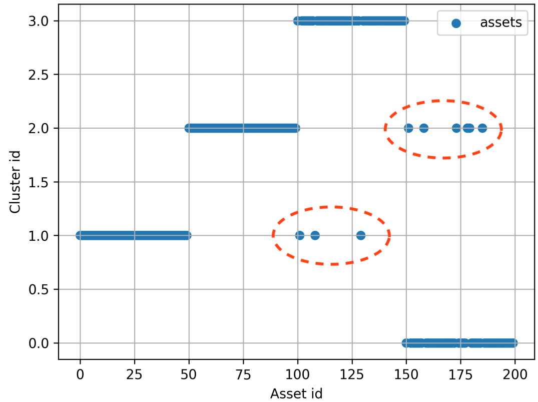
Figure 4. An example ROC for asset id 52 evaluated for testing dataset with l and L equal to 0 and 10 respectively.

estimates in this scenario were evaluated according to the standard maximum likelihood estimation in (23) and (24).

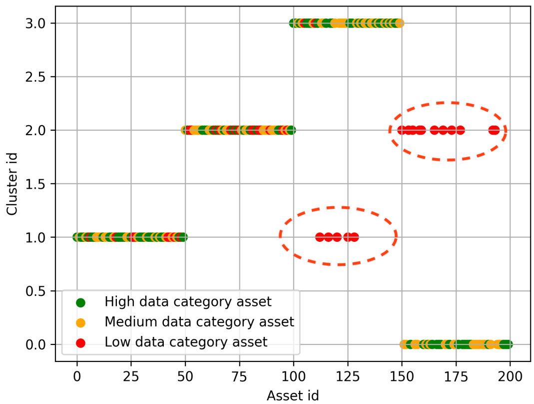
2. **Learning from similar assets** In this scenario, the hierarchical model for the fleet was implemented. Clusters of similar assets were identified, and the parameters for the hierarchical model were estimated using the EM algorithm as explained in Section 3. The EM steps were iterated 20 times, and 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.



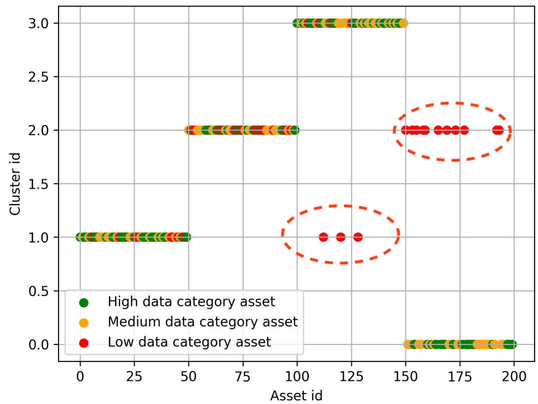
(a) All assets have 5 data points each



(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

Figure 5. 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..

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. 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 5, 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 4.1.3. 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.

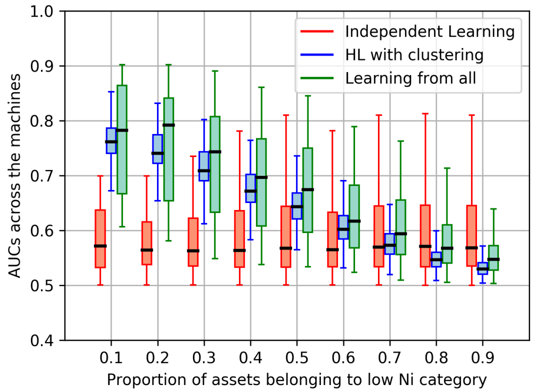
In the real world, this problem can be addressed by including certain categorical data along with the time series data. Categorical data can arise from the operational experience, such as asset's environment, upkeep, operation, etc. However, for the experimental results presented here, it was assured 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.

4.4. Experimental Results

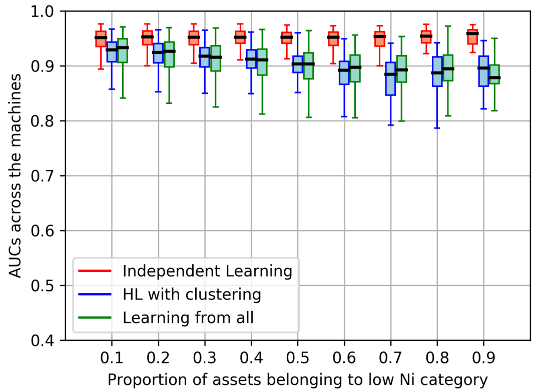
For each of the four scenarios, the AUCs were evaluated across all assets in the fleet. Box plots for each low, medium, and high data category assets for the same testing dataset are shown in Figure 6, where “HL" stands for “Hierarchical Learning" where the final estimates are estimated based on the higher level model. Figure 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 in steps of 1, 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. Figures 7 and 8 present 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 Figure captions.

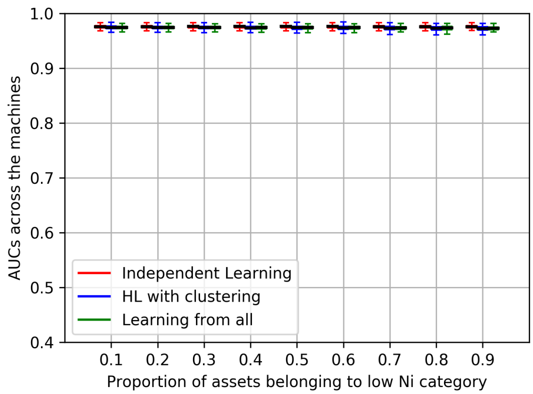
Furthermore, a learning scenario where all 800 assets held the same amount of data was also studied. This was done by simulating the fleet where all assets initially had 5 data points only, which were gradually increased to as high as 500 together across all assets. The classifier performances were studied throughout this transition. Figure 9 present the classifier performances when all assets contained the same amount of data.



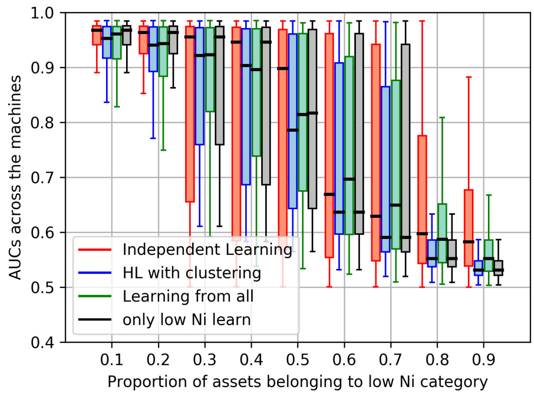
(a) AUCs across low data category assets only.



(b) AUCs across medium data category assets only.

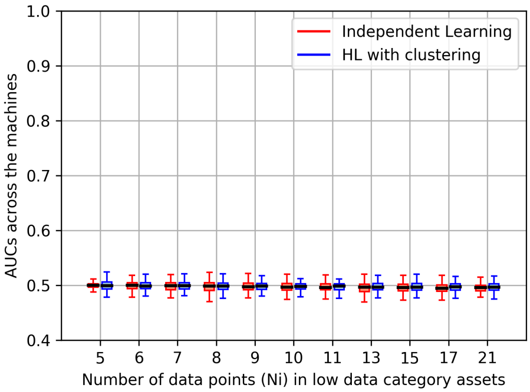


(c) AUCs across high data category assets only.

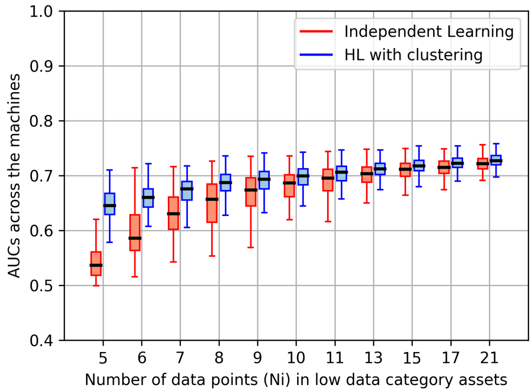


(d) AUCs across all assets, and for all the cases included in the experiments.

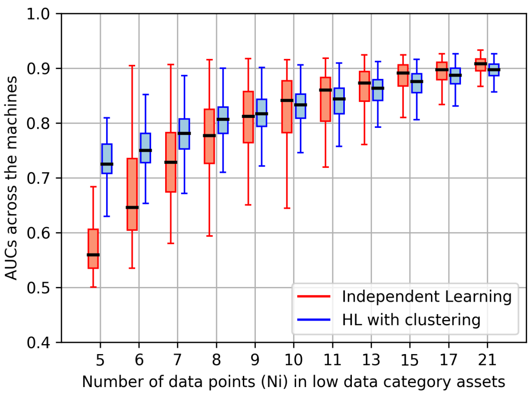
Figure 6. Shown here are the AUCs measured for the experiment cases. The subset of assets across which the AUCs are measured are indicated in the corresponding captions. For all the above four plots, the deviation for anomalous data in the testing dataset was set at 1 and 10 for l and L respectively.



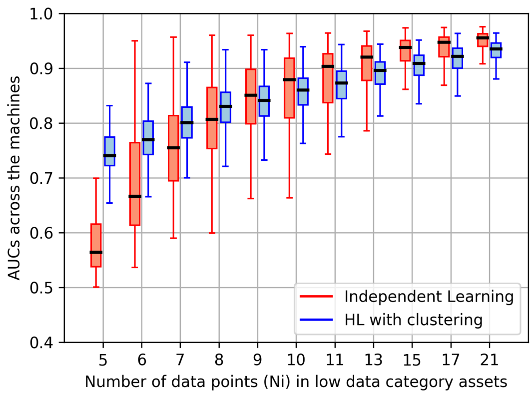
(a) $l = 0; L = 1$



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

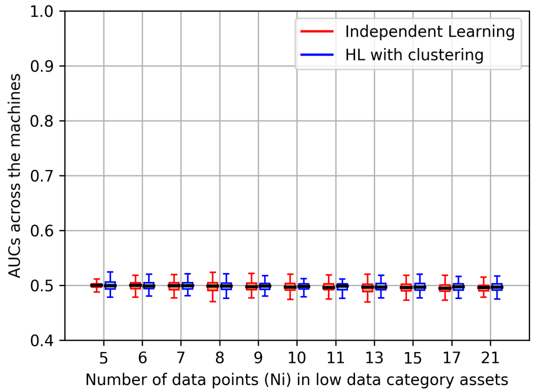


(c) $l = 0; L = 5$

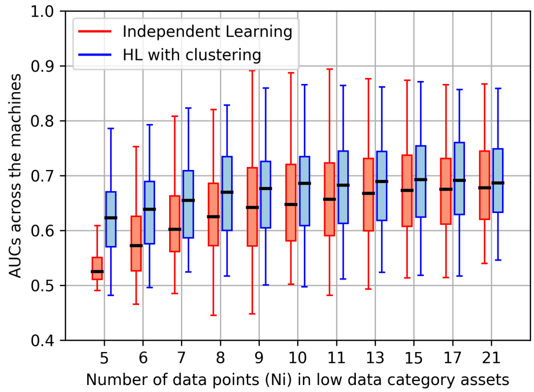


(d) $l = 0; L = 10$

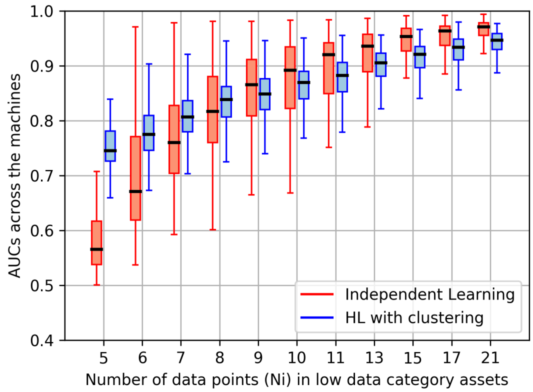
Figure 7. Box plots presenting the effect of gradually increasing data contained by the low data category assets. The captions denote the corresponding deviations in the testing dataset.



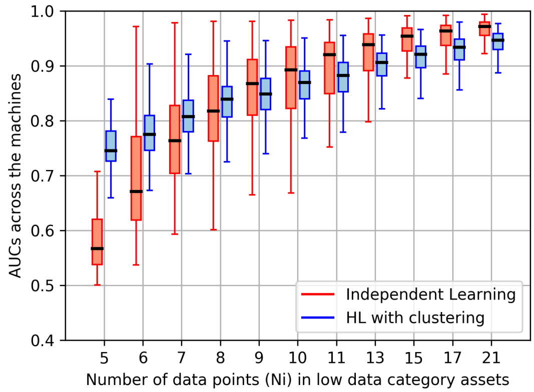
(a) $l = 0$; $L = 1$



(b) $l = 5$; $L = 1$

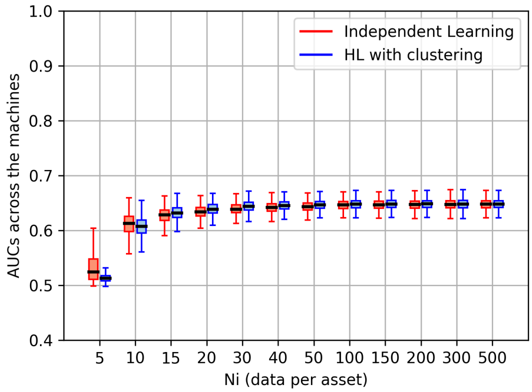


(c) $l = 20$; $L = 1$

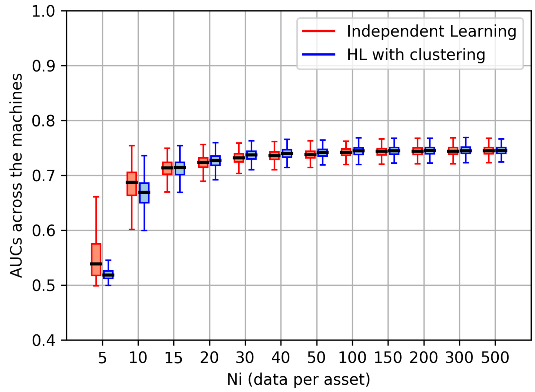


(d) $l = 100$; $L = 1$

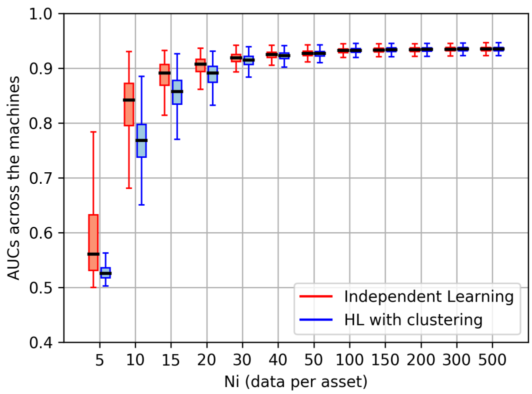
Figure 8. Box plots presenting the effect of gradually increasing data contained by the low data category assets. The captions denote the corresponding deviations in the testing dataset.



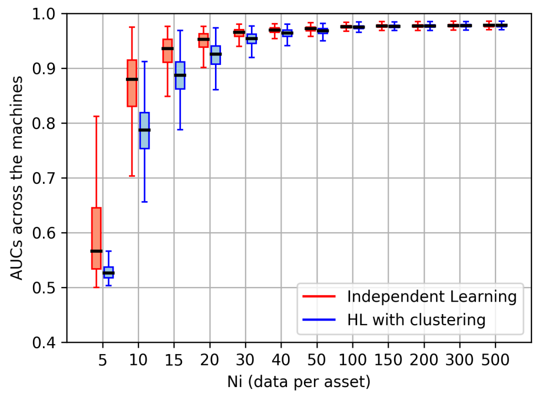
(a) $l = 0$; $L = 1.5$



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



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



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

Figure 9. 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.

5. Discussion

A better classifier for a given asset and a testing dataset is characterised by a higher AUC, as explained in Section 4.2. 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. A classifier's consistency is represented as variance of AUCs measured across the assets by the length of the boxes and whiskers of the box plots. Therefore, while analysing the results, one must note that a classifier showing higher median and smaller variance is a better classifier. The following points are summarised from the results presented in Section 4.4:

1. It is inferred from Figures 6a to 6c that hierarchical modelling is beneficial for the assets belonging to the low data category only. Medium and High data category assets are better off or equally good learning from their own data rather than learning from others. For the assets belonging to the low data category however, 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. Figures 7 and 8 also show that until a certain amount of data is accumulated by the asset, it is better for it to rely on hierarchical model estimates. While transition in figures 7 and 8 occurs at

approximately 13 data points, it can be different for different applications depending on the asset similarities, variance in data, and the proportion of low data category assets present in the fleet.

2. Figure 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 6a to 6c. Figure 6d also shows that the AUCs are higher when only the low data assets rely on the hierarchical model estimates, while others learn from their own data using independent models.

   The aforementioned points are further highlighted by Figures 10 and 11 (in Appendix C) 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.

3. Unfortunately, Figure 9 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 9 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 9 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.
4. Apart from the performance evaluation metric presented in Section 4.2, Bhattacharyya distance (D_B) was also used to compare the performances of hierarchical and independent asset models. D_B is a metric that evaluates the similarity between two multivariate Gaussians, and is calculated according to (28) for the Gaussians parameterised by $(\boldsymbol{\mu}_1, \mathbf{C}_1)$ and $(\boldsymbol{\mu}_2, \mathbf{C}_2)$ (Bhattacharyya, 1946). For the current application, D_B between the true and estimated Gaussians for all the assets were evaluated. The plots for the evaluated D_B are presented in Appendix D.

$$D_B = \frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \left(\frac{\mathbf{C}_1 + \mathbf{C}_2}{2}\right)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2} ln \left(\frac{\text{Det}\left(\frac{\mathbf{C}_1 + \mathbf{C}_2}{2}\right)}{\sqrt{\text{Det}(\mathbf{C}_1)\text{Det}(\mathbf{C}_2)}} \right) \tag{28}$$

5. It was observed that the performance of hierarchical model was affected by the choice of range of means mentioned in 4.1.3. Had the range of means been shorter, it would mean 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 E and D for both performance metrics.

6. Conclusion

This paper proposes the use of hierarchical model as a systematic method 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.

The experiments demonstrate collaborative learning using hierarchical model, showing that it can significantly improve the performances of conventional classifiers in the early periods of asset operations, when sufficient data are not available to estimate the Gaussian parameters using conventional maximum likelihood methods. However, the hierarchical model performs worse than maximum likelihood estimation when all assets in the fleet have same amount of data. The higher level distributions are also representative of the general behaviour of the asset fleet, and can be of interest to the operators who want an overall understanding the fleet performance.

7. Future Research Directions

This was the first use case of hierarchical modelling for anomaly detection in industrial asset operations, and interesting future research awaits.

1. The example implementation here was shown using a simulation fleet of assets. An interesting follow up work would be to analyse how hierarchical modelling works for a real world fleet of assets. Such analysis can include the extent of improvement in overall maintenance cost to the organisation, and therefore its business value can be justified. 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 4.3.
2. Since anomaly detection algorithms are supposed to be implemented in real time, a follow up task is to extend the hierarchical model to an online version. The online version should classify each new data point as anomalous or not, and if the new data point is not anomalous it should be used to update the hierarchical model parameters.
3. An important conclusion from the experiments 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.
4. Lastly, an important assumption while modelling the asset behaviours was that the mean of the Gaussian during an asset's normal operation is constant. This might not always be the case. Sometimes, an asset's operation could involve a sequence of tasks which could induce a cyclic nature to the Gaussian mean. Therefore, future research must focus on extending the current hierarchical model to account for natural deviations observed in the Gaussian mean throughout an asset's operation.

Acknowledgments. We are grateful for the discussions and feedbacks from Manuel Herrera, Marco Perez Hernandez, and Amit Kumar Jain from the Institute for Manufacturing, Cambridge CB3 0FS, UK.

Funding statement. This research was funded by the EPSRC and BT Prosperity Partnership project: Next Generation Converged Digital Infrastructure, grant number EP/R004935/1. This research was also funded by Siemens Industrial Turbomachinery, Lincoln, UK LN5 7FD.

Competing interests. 'None'

Data availability statement. The data used to generate the results presented in this paper can be found in: https://github.com/Dhada27/Hierarchical-Modelling-Asset-Fleets

Ethical standards. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

Author contributions. Conceptualisation: Maharshi Dhada; Mark Girolami; Ajith Kumar Parlikad. Methodology: Mark Girolami; Maharshi Dhada. Data curation: Mark Girolami; Maharshi Dhada; Ajith Kumar Parlikad. Data visualisation: Maharshi Dhada; Ajith Kumar Parlikad. Writing original draft: Maharshi Dhada. All authors approved the final submitted draft.

References

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.

Arjas, E. and Bhattacharjee, M. (2004). Modelling Heterogeneity in Repeated Failure Time Data: A Hierarchical Bayesian Approach. In *Mathematical Reliability: An Expository Perspective*, pages 71–86. Springer, Boston, MA.

Ascher, H. (1983). Disccussion of "Point Procecsses and Renewal Theory: A Brief Survey". In *NATO ASI Series, Series F: Computer and Systems Sciences*.

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

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).

Chen, J. and Singpurwalla, N. D. (1996). The Notion of "Composite Reliability" and Its Hierarchical Bayes Estimation. *Journal of the American Statistical Association*, 91(436):1474.

Dedecius, K. and Ettler, P. (2014). Hierarchical modelling of industrial system reliability with probabilistic logic. In *ICINCO 2014 - Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics*, volume 1.

Eckert, N., Parent, E., Bélanger, L., and Garcia, S. (2007). Hierarchical Bayesian modelling for spatial analysis of the number of avalanche occurrences at the scale of the township. *Cold Regions Science and Technology*, 50(1-3):97–112.

Economou, T., Kapelan, Z., and Bailey, T. (2007). An aggregated hierarchical Bayesian model for the prediction of pipe failures. In *Proceedings of the Combined International Conference of Computing and Control for the Water Industry, CCWI2007 and Sustainable Urban Water Management, SUWM2007*.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

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

Gilchrist, A. and Gilchrist, A. (2016). Smart Factories. In *Industry 4.0*, pages 217–230. Apress.

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.

Hensman, J., Lawrence, N. D., and Rattray, M. (2013). Hierarchical Bayesian modelling of gene expression time series across irregularly sampled replicates and clusters. *BMC Bioinformatics*, 14(1):252.

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.

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.

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.

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

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

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.

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.

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

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.

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.

Lindley, D. V., Cox, D. R., and Lewis, P. A. W. (1967). The Statistical Analysis of Series of Events. *The Mathematical Gazette*, 51(377).

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

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

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

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.
Salvador Palau, A., Liang, Z., Lütgehetmann, D., and Parlikad, A. K. (2019). Collaborative prognostics in Social Asset Networks. *Future Generation Computer Systems*, 92:987–995.
Saxena, A., Goebel, K., Simon, D., and Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management, PHM 2008*.
Teacy, W. T., Luck, M., Rogers, A., and Jennings, N. R. (2012). An efficient and versatile approach to trust and reputation using hierarchical Bayesian modelling. *Artificial Intelligence*, 193:149–185.
Thill, M. (2017). The Relationship between the Mahalanobis Distance and the Chi-Squared Distribution.
Xu, L. D., He, W., and Li, S. (2014). Internet of things in industries: A survey.
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.
Yuan, T. and Ji, Y. (2015). A hierarchical bayesian degradation model for heterogeneous data. *IEEE Transactions on Reliability*, 64(1).
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.
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.

A. Derivations of the E and M steps

A.1. 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 $\boldsymbol{\theta} = \boldsymbol{\theta}^t$. Since $\mathbf{z}_{i,k}$ is binary, $\mathbb{E}(\mathbf{z}_{i,k}|(\boldsymbol{\mu}_i, \mathbf{C}_i)^t, \boldsymbol{\theta}^t) = p(\mathbf{z}_{i,k} = 1|(\boldsymbol{\mu}_i, \mathbf{C}_i)^t, \boldsymbol{\theta}^t) = p(\mathbf{z}_{i,k} = 1|(\boldsymbol{\mu}_i, \mathbf{C}_i)^t, \boldsymbol{\theta}^t)$. Using Bayes' rule:

$$p(\mathbf{z}_{i,k} = 1|(\boldsymbol{\mu}_i, \mathbf{C}_i)^t, \boldsymbol{\theta}^t) = \frac{p((\boldsymbol{\mu}_i, \mathbf{C}_i)^t|\mathbf{z}_{i,k} = 1, \boldsymbol{\theta}^t)p(\mathbf{z}_{i,k} = 1)}{\sum_{k=1}^{K} p((\boldsymbol{\mu}_i, \mathbf{C}_i)^t|\mathbf{z}_{i,k} = 1, \boldsymbol{\theta}^t)p(\mathbf{z}_{i,k} = 1)} \tag{29}$$

from equations 8 and 10 we know,

$$p(\mathbf{z}_{i,k} = 1|(\boldsymbol{\mu}_i, \mathbf{C}_i)^t, \boldsymbol{\theta}^t) = \frac{(\mathcal{N}(\boldsymbol{\mu}_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k))(\pi_k)}{\sum_{k=1}^{K}(\mathcal{N}(\boldsymbol{\mu}_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k))(\pi_k)} \tag{30}$$

Where all distribution parameters correspond to the values obtained at M-step of latest (t^{th}) iteration. Let, $p(\mathbf{z}_{i,k} = 1|(\boldsymbol{\mu}_i, \mathbf{C}_i)^t, \boldsymbol{\theta}^t) = \boldsymbol{\gamma}_{i,k}$. Therefore, our function $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t)$ can be deduced from equation 14 by replacing $\mathbf{z}_{i,k}$ with $\boldsymbol{\gamma}_{i,k}$:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) = \sum_{i=1}^{I}\sum_{n=1}^{N_i} \log(\mathcal{N}(\boldsymbol{\mu}_i, \mathbf{C}_i)) + \sum_{i=1}^{I}\sum_{k=1}^{K} \boldsymbol{\gamma}_{i,k} \log\left(\pi_k \mathcal{N}(\boldsymbol{\mu}_i|\mathbf{m}_k, \beta_k^{-1}\mathbf{C}_i)\mathcal{IW}(\mathbf{C}_i|\boldsymbol{\Lambda}_k, \alpha_k)\right) \tag{31}$$

After substituting the symbolic representation with the corresponding distribution functions and parameters, $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t)$ (not including constant terms, because they would become zero after differentiation) becomes:

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) = \\
-\frac{1}{2}\sum_i\sum_n log|\mathbf{C}_i| - \frac{1}{2}\sum_i\sum_n(\mathbf{x}_{i,n} - \boldsymbol{\mu}_i)^T\mathbf{C}_i^{-1}(\mathbf{x}_{i,n} - \boldsymbol{\mu}_i) - \\
\frac{1}{2}\sum_i \boldsymbol{\gamma}_{i,k}\sum_k \log|\mathbf{C}_i| + \frac{1}{2}\sum_i \boldsymbol{\gamma}_{i,k}\sum_k \log(\beta_k) - \frac{1}{2}\sum_i \boldsymbol{\gamma}_{i,k}\sum_k \beta_k(\boldsymbol{\mu}_i - \mathbf{m}_k)^T\mathbf{C}_i^{-1}(\boldsymbol{\mu}_i - \mathbf{m}_k) + \\
\frac{1}{2}\sum_i \boldsymbol{\gamma}_{i,k}\sum_k \alpha_k \log|\boldsymbol{\Lambda}_k| - \frac{1}{2}\sum_i \boldsymbol{\gamma}_{i,k}\sum_k \alpha_k d\log(2) - \\
\sum_i \boldsymbol{\gamma}_{i,k}\sum_k \log\left(\Gamma_d(\frac{\alpha_k}{2})\right) - \frac{1}{2}\sum_i \boldsymbol{\gamma}_{i,k}\sum_k(\alpha_k + d + 1)\log|\mathbf{C}_i| - \\
\frac{1}{2}\sum_i \boldsymbol{\gamma}_{i,k}\sum_k Tr(\boldsymbol{\Lambda}_k\mathbf{C}_i^{-1}) + \sum_i \boldsymbol{\gamma}_{i,k}\sum_k \pi_k
\end{aligned} \tag{32}
$$

The $\boldsymbol{\gamma}_{i,k}$ are not included in summations because they are supposed to be treated as constants in the M-step that follows.

A.2. M-step

In M-step, $\boldsymbol{\theta}^{t+1}$ values are obtained for following $(t+1)^{th}$ E-step by maximising the $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t)$ function obtained in equation 32 with respect to each of the $\boldsymbol{\theta}$ parameters, and treating $\boldsymbol{\gamma}_{i,k}$ as constants. Calculations for partial derivatives of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t)$ w.r.t. each parameter are shown below:

A.2.1. Evaluating $\hat{\boldsymbol{\mu}}_i$

$$
\begin{aligned}
\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t)}{\partial \boldsymbol{\mu}_i} &\implies \sum_n \mathbf{C}_i^{-1}(\mathbf{x}_{i,n} - \boldsymbol{\mu}_i) - \sum_k \beta_k\boldsymbol{\gamma}_{i,k}\mathbf{C}_i^{-1}(\boldsymbol{\mu}_i - \mathbf{m}_k) = 0 \\
&\implies \sum_n \mathbf{x}_{i,n} - N_i\boldsymbol{\mu}_i = \boldsymbol{\mu}_i\sum_k \beta_k\boldsymbol{\gamma}_{i,k} - \sum_k \beta_k\boldsymbol{\gamma}_{i,k}\mathbf{m}_k \\
&\implies \hat{\boldsymbol{\mu}}_i = \frac{1}{N_i + \sum_{k=1}^K \beta_k\boldsymbol{\gamma}_{i,k}}\left[\sum_{n=1}^{N_i}\mathbf{x}_{i,n} + \sum_{k=1}^K \beta_k\boldsymbol{\gamma}_{i,k}\mathbf{m}_k\right]
\end{aligned}
$$

A.2.2. Evaluating $\hat{\mathbf{m}}_k$

$$
\begin{aligned}
\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t)}{\partial \mathbf{m}_k} &\implies \sum_i \boldsymbol{\gamma}_{i,k}\beta_k\mathbf{C}_i^{-1}(\boldsymbol{\mu}_i - \mathbf{m}_k) = 0 \\
&\implies \beta_k\sum_i \boldsymbol{\gamma}_{i,k}\mathbf{C}_i^{-1}\boldsymbol{\mu}_i = \beta_k\sum_i \boldsymbol{\gamma}_{i,k}\mathbf{C}_i^{-1}\mathbf{m}_k \\
&\implies \left[\sum_i \boldsymbol{\gamma}_{i,k}\mathbf{C}_i^{-1}\right]\mathbf{m}_k = \left[\sum_i \boldsymbol{\gamma}_{i,k}\mathbf{C}_i^{-1}\boldsymbol{\mu}_i\right] \\
&\implies \hat{\mathbf{m}}_k = \left[\sum_{i=1}^I \boldsymbol{\gamma}_{i,k}\mathbf{C}_i^{-1}\right]^{-1}\left[\sum_{i=1}^I \boldsymbol{\gamma}_{i,k}\mathbf{C}_i^{-1}\boldsymbol{\mu}_i\right]
\end{aligned}
$$

A.2.3. Evaluating $\hat{\mathbf{\Lambda}}_k$

$$
\begin{aligned}
\frac{\partial Q(\boldsymbol{\theta},\boldsymbol{\theta}^t)}{\partial \mathbf{\Lambda}_k} &\implies \frac{1}{2}\sum_i \boldsymbol{\gamma}_{i,k}\alpha_k \mathbf{\Lambda}_k^{-1} - \frac{1}{2}\sum_i \boldsymbol{\gamma}_{i,k}\mathbf{C}_i^{-1} = 0 \\
&\implies \mathbf{\Lambda}_k^{-1} = \frac{\sum_i \boldsymbol{\gamma}_{i,k}\mathbf{C}_i^{-1}}{\alpha_k \sum_i \boldsymbol{\gamma}_{i,k}} \\
&\implies \hat{\mathbf{\Lambda}}_k = \left[\alpha_k \sum_{i=1}^{I} \boldsymbol{\gamma}_{i,k}\right]\left[\sum_{i=1}^{I} \boldsymbol{\gamma}_{i,k}\mathbf{C}_i^{-1}\right]^{-1}
\end{aligned}
$$

A.2.4. Evaluating $\hat{\beta}_k$

$$
\begin{aligned}
\frac{\partial Q(\boldsymbol{\theta},\boldsymbol{\theta}^t)}{\partial \beta_k} &\implies \frac{d}{2}\sum_i \boldsymbol{\gamma}_{i,k}\frac{1}{\beta_k} - \frac{1}{2}\sum_i \boldsymbol{\gamma}_{i,k}(\boldsymbol{\mu}_i - \mathbf{m}_k)^T \mathbf{C}_i^{-1}(\boldsymbol{\mu}_i - \mathbf{m}_k) = 0 \\
&\implies \frac{1}{\hat{\beta}_k} = \frac{\sum_{i=1}^{I} \boldsymbol{\gamma}_{i,k}(\boldsymbol{\mu}_i - \mathbf{m}_k)^T \mathbf{C}_i^{-1}(\boldsymbol{\mu}_i - \mathbf{m}_k)}{d\sum_{i=1}^{I} \boldsymbol{\gamma}_{i,k}}
\end{aligned}
$$

A.2.5. Evaluating $\hat{\mathbf{C}}_i$

$$
\begin{aligned}
\frac{\partial Q(\boldsymbol{\theta},\boldsymbol{\theta}^t)}{\partial \mathbf{C}_i} &\implies -\frac{N_i}{2}\mathbf{C}_i^{-1} + \frac{1}{2}\mathbf{C}_i^{-1}\Big(\sum_n (\mathbf{x}_{i,n} - \boldsymbol{\mu}_i)(\mathbf{x}_{i,n} - \boldsymbol{\mu}_i)^T\Big)\mathbf{C}_i^{-1} \\
&- \frac{1}{2}\mathbf{C}_i^{-1} + \frac{1}{2}\mathbf{C}_i^{-1}\Big(\sum_k \beta_k \boldsymbol{\gamma}_{i,k}(\boldsymbol{\mu}_i - \mathbf{m}_k)(\boldsymbol{\mu}_i - \mathbf{m}_k)^T\Big)\mathbf{C}_i^{-1} \\
&- \frac{1}{2}\sum_k \boldsymbol{\gamma}_{i,k}(\alpha_k + d + 1)\mathbf{C}_i^{-1} + \frac{1}{2}\sum_k \boldsymbol{\gamma}_{i,k}\mathbf{C}_i^{-1}\mathbf{\Lambda}_k\mathbf{C}_i^{-1} = 0 \\
&\implies -\frac{N_i}{2}\mathbf{C}_i + \frac{1}{2}\sum_n (\mathbf{x}_{i,n} - \boldsymbol{\mu}_i)(\mathbf{x}_{i,n} - \boldsymbol{\mu}_i)^T - \frac{1}{2}\mathbf{C}_i \\
&+ \frac{1}{2}\sum_k \beta_k \boldsymbol{\gamma}_{i,k}(\boldsymbol{\mu}_i - \mathbf{m}_k)(\boldsymbol{\mu}_i - \mathbf{m}_k)^T \\
&- \frac{1}{2}\sum_k \boldsymbol{\gamma}_{i,k}(\alpha_k + d + 1)\mathbf{C}_i + \frac{1}{2}\sum_k \boldsymbol{\gamma}_{i,k}\mathbf{\Lambda}_k = 0 \\
&\implies (N_i + 1 + \sum_k \boldsymbol{\gamma}_{i,k}\alpha_k + d + 1)\mathbf{C}_i = \sum_n (\mathbf{x}_{i,n} - \boldsymbol{\mu}_i)(\mathbf{x}_{i,n} - \boldsymbol{\mu}_i)^T \\
&+ \sum_k \beta_k \boldsymbol{\gamma}_{i,k}(\boldsymbol{\mu}_i - \mathbf{m}_k)(\boldsymbol{\mu}_i - \mathbf{m}_k)^T + \sum_k \boldsymbol{\gamma}_{i,k}\mathbf{\Lambda}_k \\
&\implies \hat{\mathbf{C}}_i = \\
&\frac{\sum_{n=1}^{N_i} (\mathbf{x}_{i,n} - \boldsymbol{\mu}_i)(\mathbf{x}_{i,n} - \boldsymbol{\mu}_i)^T + \sum_{k=1}^{K} \beta_k \boldsymbol{\gamma}_{i,k}(\boldsymbol{\mu}_i - \mathbf{m}_k)(\boldsymbol{\mu}_i - \mathbf{m}_k)^T + \sum_{k=1}^{K} \boldsymbol{\gamma}_{i,k}\mathbf{\Lambda}_k}{N_i + \sum_{k=1}^{K} \boldsymbol{\gamma}_{i,k}\alpha_k + d + 2}
\end{aligned}
$$

A.2.6. Evaluating $\hat{\alpha}_k$

The below stated $f(\alpha_k)$ must be maximised w.r.t. α_k:

$$f(\alpha_k) = \frac{1}{2}\alpha_k \log|\mathbf{\Lambda}_k| \sum_i \boldsymbol{\gamma}_{ik} - \frac{d}{2}\log(2)\alpha_k \sum_i \boldsymbol{\gamma}_{ik} - \log\left(\Gamma_d\left(\frac{\alpha_k}{2}\right)\right)\sum_i \boldsymbol{\gamma}_{ik} - \frac{1}{2}(\alpha_k + d + 1)\sum_i \boldsymbol{\gamma}_{ik} \log|\boldsymbol{C}_i| \quad (33)$$

But the presence of $\log\left(\Gamma_d\left(\frac{\alpha_k}{2}\right)\right)\sum_i \boldsymbol{\gamma}_{ik}$ 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 $\alpha_k \in (d, d+20)$.

A.2.7. Evaluating $\hat{\boldsymbol{\pi}}_k$

Evaluating $\hat{\boldsymbol{\pi}}_k$ 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 $[Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) + \eta(\sum_k \pi_k - 1)]$ w.r.t. π_k, where η is the Lagrange multiplier. From equation 32, we have:

$$\frac{\partial[Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) + \eta(\sum_k \pi_k - 1)]}{\partial \pi_k} \implies \frac{\sum_i \boldsymbol{\gamma}_{i,k}}{\pi_k} + \eta = 0$$
$$\implies \pi_k = \frac{-\sum_i \boldsymbol{\gamma}_{i,k}}{\eta}$$

But since $\sum_k \pi_k = 1$; $\eta = \eta(\sum_k \pi_k) = -\sum_i \sum_k \boldsymbol{\gamma}_{i,k}$ (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:

$$\hat{\boldsymbol{\pi}}_k = \frac{\sum_{i=1}^{I} \boldsymbol{\gamma}_{i,k}}{I}$$

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 (Thill, 2017) 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 $\mathbf{\Sigma}$ is evaluated as (assuming orthonormal eigenvectors):

$$D^2_{md} = (X - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(X - \boldsymbol{\mu})$$

Upon performing he eigenvalue decomposition of $\mathbf{\Sigma}^{-1}$, one obtains:

$$\mathbf{\Sigma}^{-1} = U\Lambda^{-1}U^{-1} = U\Lambda U^T = \sum_{k=1}^{d} \lambda_k^{-1} u_k u_k^T$$

Where u_k is the k^{th} eigenvector of the corresponding eigenvalue λ_k.

Therefore,

$$
\begin{aligned}
D_{md} &= (X-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(X-\boldsymbol{\mu}) && (34)\\
&= (X-\boldsymbol{\mu})^T \Big(\sum_{k=1}^{d} \lambda_k^{-1} u_k u_k^T\Big)(X-\boldsymbol{\mu}) && (35)\\
&= \sum_{k=1}^{d} \lambda_k^{-1} (X-\boldsymbol{\mu})^T u_k u_k^T (X-\boldsymbol{\mu}) && (36)\\
&= \sum_{k=1}^{d} \lambda_k^{-1} \Big[\mu_k^T (X-\boldsymbol{\mu})\Big]^2 && (37)\\
&= \sum_{k=1}^{d} \Big[\lambda_k^{\frac{-1}{2}} \mu_k^T (X-\boldsymbol{\mu})\Big] && (38)\\
&= \sum_{k=1}^{d} Y_k^2 && (39)
\end{aligned}
$$

Where Y_k is a new random variable based on affine linear transformation of the random vector X.

We know that a random variable $Z = (X - \boldsymbol{\mu})$ can be expressed as $Z \sim \mathcal{N}(0, \boldsymbol{\Sigma})$. 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 $Y_k \sim \mathcal{N}(0, \boldsymbol{\Sigma}_k^2)$ where:

$$
\begin{aligned}
\boldsymbol{\Sigma}_k^2 &= \lambda_k^{\frac{-1}{2}} u_k^T \boldsymbol{\Sigma} \lambda_k^{\frac{-1}{2}} u_k\\
&= \lambda_k^{-1} u_k^T \boldsymbol{\Sigma} \mu_k
\end{aligned}
$$

Upon substituting $\boldsymbol{\Sigma} = \sum_{j=1}^{d} \lambda_j u_j u_j^T$,

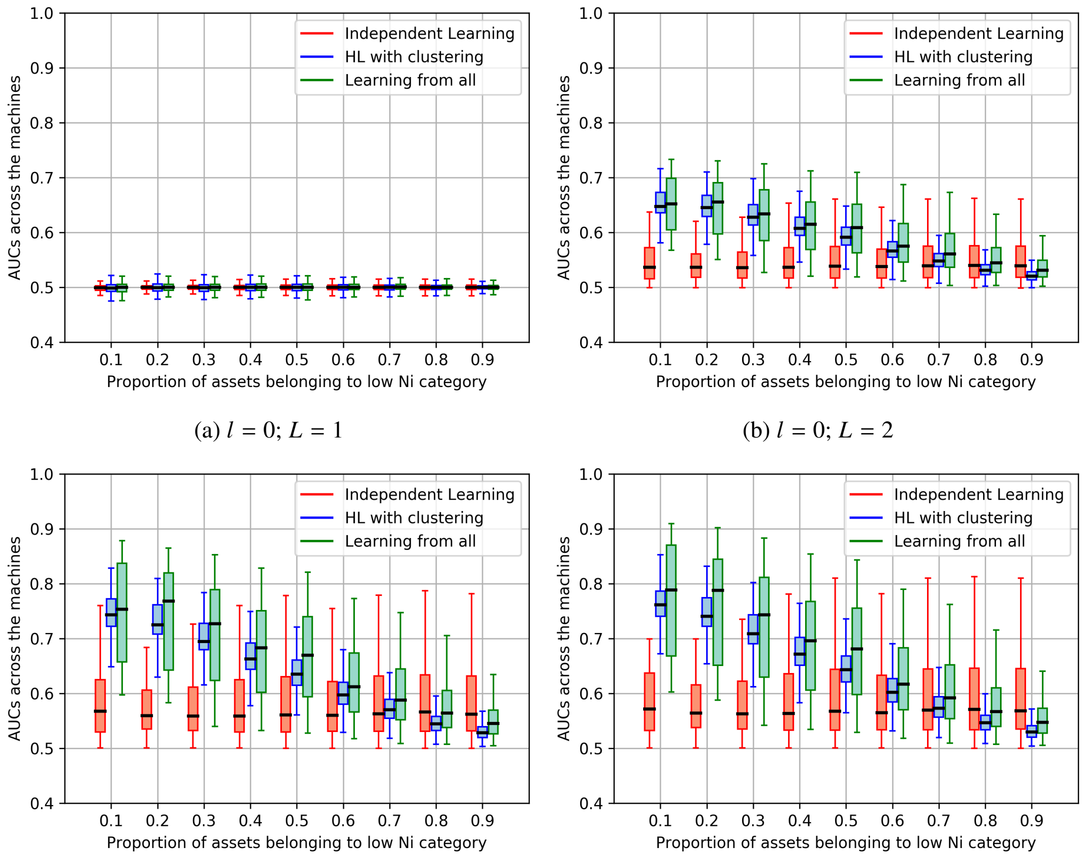
$$
\begin{aligned}
\boldsymbol{\Sigma}_k^2 &= \lambda_k^{-1} u_k^T \boldsymbol{\Sigma} \mu_k\\
&= \lambda_k^{-1} u_k^T \Big(\sum_{j=1}^{d} \lambda_j u_j u_j^T\Big) \boldsymbol{\mu}_k\\
&= \sum_{j=1}^{d} \lambda_k^{-1} u_k^T \lambda_j u_j u_j^T u_k\\
&= \sum_{j=1}^{d} \lambda_k^{-1} \lambda_j u_k^T u_j u_j^T u_k
\end{aligned}
$$

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.

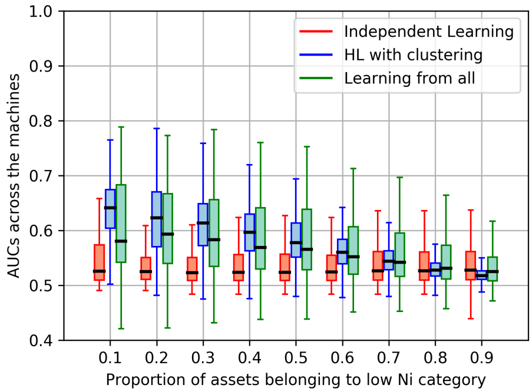
C. More result figures to demonstrate the benefit of hierarchical modelling for the low data category assets.



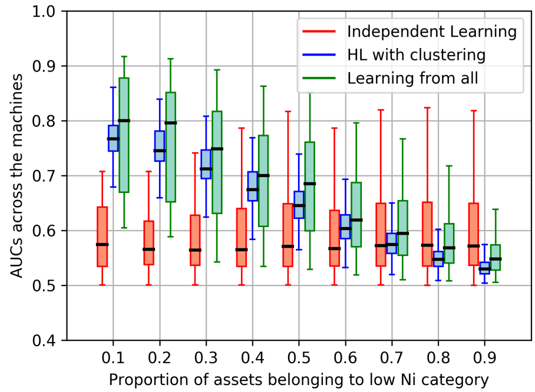
(a) $l = 0$; $L = 1$ (b) $l = 0$; $L = 2$

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

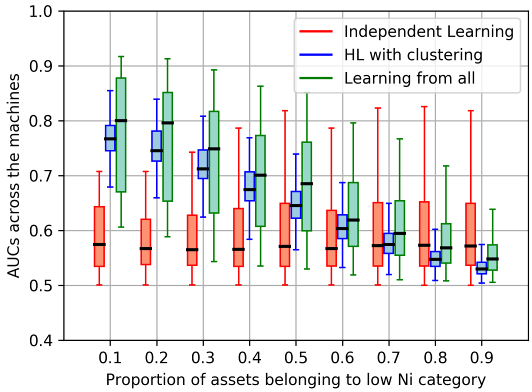
Figure 10. Box plots presenting AUCs recorded across the assets belonging to the low data category. The corresponding testing dataset deviations are denoted in the captions.



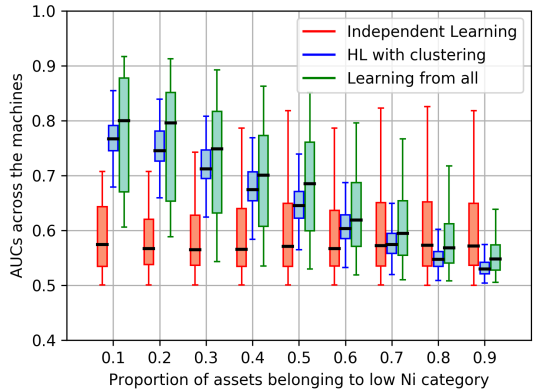
(a) $l = 5$; $L = 1$



(b) $l = 20$; $L = 1$



(c) $l = 50$; $L = 1$

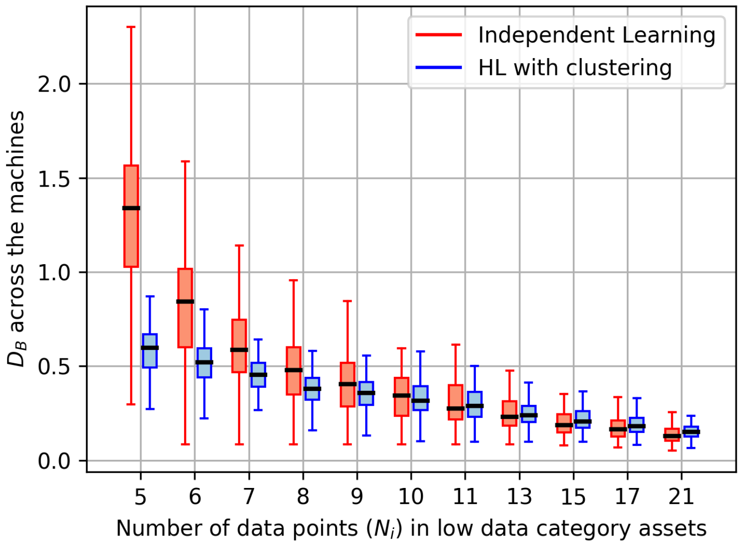


(d) $l = 100$; $L = 1$

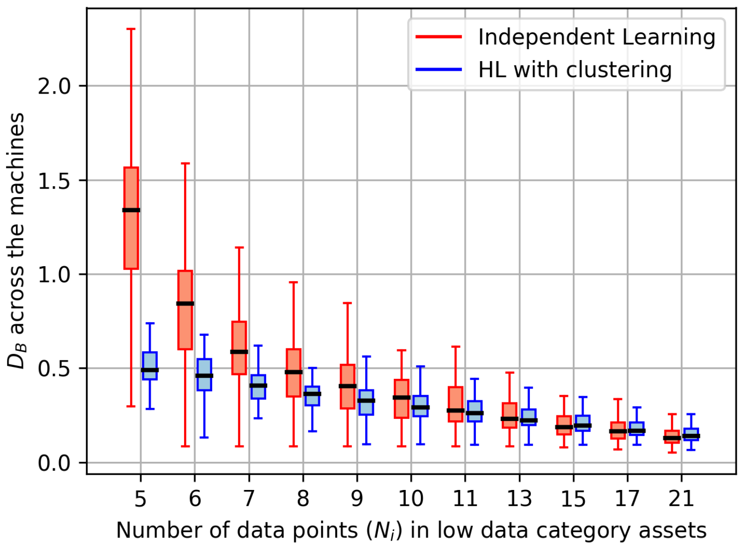
Figure 11. Box plots presenting AUCs recorded across the assets belonging to the low data category. The corresponding testing dataset deviations are denoted in the captions.

D. Performance evaluation using Bhattacharyya distance

Figures 12a and 12b present the Bhattacharyya distance (D_B) evaluated across all assets in the fleet, according to (28), as the data points in the low data category assets were sequentially increased. Figure 12a 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 12b 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 (25).



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



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

Figure 12. Box plots presenting the D_B recorded across the assets belonging to the low data category, for the original and narrower range of means.

E. Results from the experiment conducted for a shorter range of asset means.

Figure 13 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 (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 13 is evaluated in the same manner as Figure 8 but for the training and testing datasets corresponding to a narrower range of means.

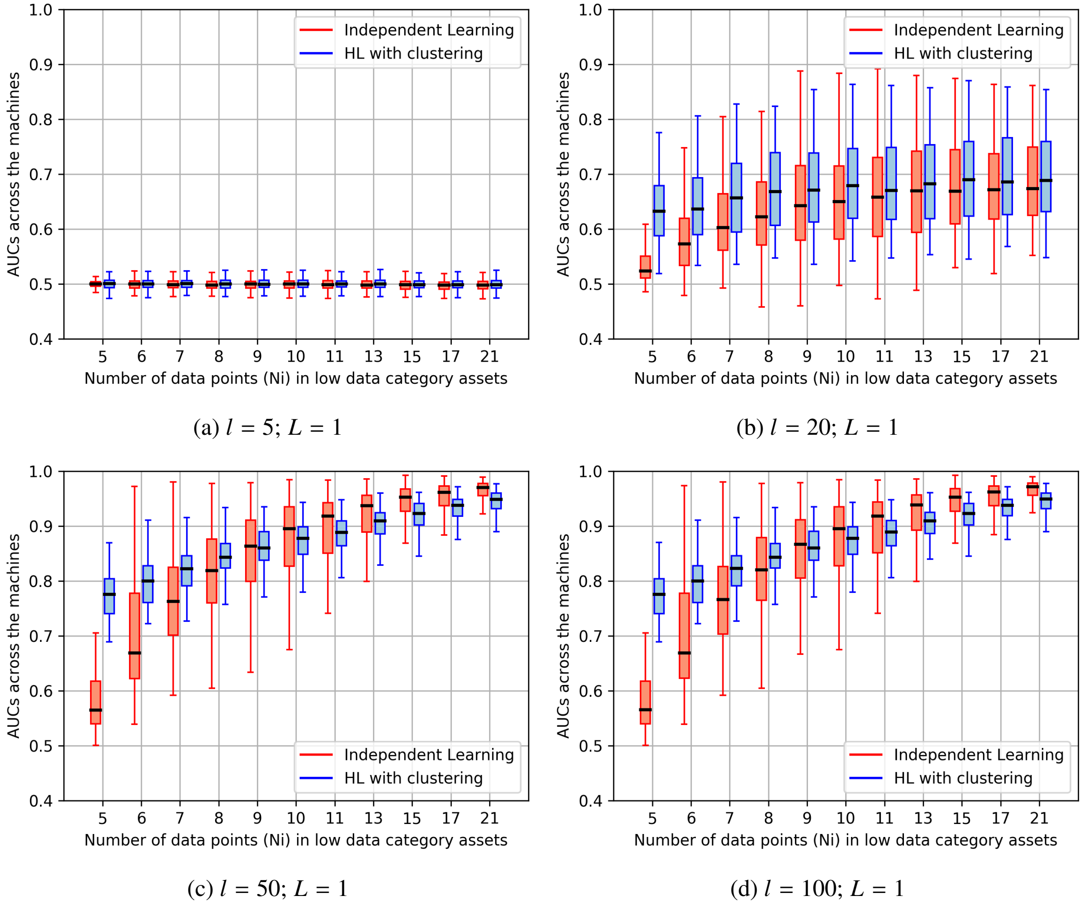


Figure 13. Box plots presenting AUCs recorded across the assets belonging to the low data category, but for a narrower range of means.