CHALMERS
UNIVERSITY OF TECHNOLOGY

Multi-Machine Gaussian Topic Modeling for Predictive Maintenance

Citation for the original published paper (version of record):
Karlsson, A., Turanoglu Bekar, E., Skoogh, A. (2021). Multi-Machine Gaussian Topic Modeling for Predictive Maintenance. IEEE Access, 9: 100063-100080.
http://dx.doi.org/10.1109/ACCESS.2021.3096387

(article starts on next page)

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 20, 2021, accepted June 23, 2021, date of publication July 12, 2021, date of current version July 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3096387

Multi-Machine Gaussian Topic Modeling for Predictive Maintenance

ALEXANDER KARLSSON[1], EBRU TURANOGLU BEKAR[2], AND ANDERS SKOOGH[2]
[1]School of Informatics, University of Skövde, 541 28 Skövde, Sweden
[2]Department of Industrial and Materials Science, Chalmers University of Technology, 412 96 Göteborg, Sweden

Corresponding author: Alexander Karlsson (alexander.karlsson@his.se)

This work was supported by Vinnova, Project: Predictive Maintenance using Advanced Cluster Analysis (PACA) under Grant 2019-00789.

ABSTRACT In this paper, we propose a coherent framework for multi-machine analysis, using a group clustering model, which can be utilized for predictive maintenance (PdM). The framework benefits from the repetitive structure posed by multiple machines and enables for assessment of health condition, degradation modeling and comparison of machines. It is based on a hierarchical probabilistic model, denoted Gaussian topic model (GTM), where cluster patterns are shared over machines and therefore it allows one to directly obtain proportions of patterns over the machines. This is then used as a basis for cross comparison between machines where identified similarities and differences can lead to important insights about their degradation behaviors. The framework is based on aggregation of data over multiple streams by a predefined set of features extracted over a time window. Moreover, the framework contains a clustering schema which takes uncertainty of cluster assignments into account and where one can specify a desirable degree of reliability of the assignments. By using a multi-machine simulation example, we highlight how the framework can be utilized in order to obtain cluster patterns and inherent variations of such patterns over machines. Furthermore, a comparative study with the commonly used Gaussian mixture model (GMM) demonstrates that GTM is able to identify inherent patterns in the data while the GMM fails. Such result is a consequence of the group level being modeled by the GTM while being absent in the GMM. Hence, the GTM are trained with a view on the data that is not available to the GMM with the consequence that the GMM can miss important, possibly even key cluster patterns. Therefore, we argue that more advanced cluster models, like the GTM, can be key for interpreting and understanding degradation behavior across machines and ultimately for obtaining more efficient and reliable PdM systems.

INDEX TERMS Exploratory data analysis, cluster analysis, Gaussian topic modeling, hierarchical modeling, multi-machine analysis, multiple data streams, predictive maintenance.

I. INTRODUCTION

The importance of manufacturing maintenance is continuously increasing in industry and academia due to the future expectations on maintenance as a key enabler of industrial digitalization. Increased levels of automation, sometimes even referred to as ''light-out factories'', can only be realized with novel maintenance solutions. As a result, the academic field has recently increased its focus towards advanced data analysis for predictive maintenance (PdM). Maintenance experts in industry and academia have therefore conceptualized smart maintenance in a digitalized manufacturing [9]. Smart maintenance highlights data-driven decision making as well as internal and external integration as important principles to improve plant performance. This means sharing and analyzing data from multiple machines to predict future maintenance activities will be central in the development of future solutions in maintenance. However, previous research has identified challenges in developing effective and precise algorithms for PdM, often due to the lack of high-quality labeled data [58], i.e., the data has not been annotated with true machine health condition or it has not contained examples of every possible fault type [23]. Correctly labeled data might not be available in real world industrial environments due to different reasons. As an example, for critical machines in a manufacturing company, the first aim is not allowing them to break down by carrying out periodic maintenance intervals. This means that no actual break-down data exists

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaojun Li.

and hence, it is difficult to define the often necessary thresholds and tolerance limits for condition monitoring. There is also no observations until the actual end of life which can be used to identify fault patterns [36], [58]. On the other hand, non-equipment-based measures such as product quality can also be used for labeling, but it is difficult to ensure that this type of information can be directly related to the corresponding machine operations. Such situations can only be managed through unsupervised machine learning methods that explore hidden structures and patterns without any target specifications [46]. Moreover, in reality, machines can generate highly heterogeneous data from multiple streams, e.g. from sensors and other computer systems, which can include a high degree of non-linearity due to the external influencing factors and other sources of uncertainties. Therefore, it is not easy to integrate all the information to provide data-driven decision support for PdM and there is a need of methods and algorithms that are able to transfer learned knowledge from one machine to another similar machine [30], [35].

We present a framework that is able to capture several of the above complexities and it allows one to perform multi-machine analysis. The framework uses an unsupervised hierarchical probabilistic model called Gaussian topic modeling (GTM) [8], [53] for exploring cluster patterns across multiple machines and where the extracted knowledge associated with identified clusters is used for machine health assessment and degradation modeling. The framework builds upon previous research [10] where one aggregates any number of features over multiple streams in order to obtain a data point that represents the state at that particular time. The main contributions are summarized as follows:

- a coherent framework where similarities and differences across machines can be utilized for learning and interpreting cluster patterns,
- a simulation procedure for exploring the above framework,
- experiments that show the benefit of utilizing the framework for exploring cluster patterns across machines.

The remainder of the paper is structured as follows. Section II focuses on related literature of PdM. The proposed framework for multi-machine analysis is described with illustrative examples in Section III. Section IV presents a multi-machine simulation; application of our framework for discovering patterns across multiple machines; results with exploration of shared clusters as well as results from a comparison with another cluster model called Gaussian mixture model (GMM). Section V discusses advantages of the proposed framework in terms of the research gaps highlighted in the introduction. Section VI concludes the paper with a summary and future work.

II. RELATED WORK

PdM has a high potential for improving productivity of manufacturing companies where the vision is failure free production with several expected effects such as: increased efficiency and reliability of operations as well as cost effectiveness. According to the estimation of Daugherty *et al.* [2], with the implementation of PdM, it is possible that companies can save up to 12% over scheduled repairs, reduce overall maintenance costs by up to 30% and eliminate asset failures up to 70%. To reach this vision and the expected effects, PdM has gained exponentially increasing attention in the context of prognostic and health management (PHM) and has been started to be adopted by many manufacturing companies in industry. Beside potential effects in industrial applications of PdM, it also holds a great interest in academia with rapidly increasing journal and conference contributions [13], [26]. In principle, PdM predicts faults and failures by real-time evaluation of the system state in production and thereby enables for efficient maintenance operations [44]. This allows for more efficient maintenance planing with reduced downtime and increased efficiency as a result [40]. Literature shows that data-driven analytics can be used to gain knowledge from data to facilitate maintenance decision-making in industrial practice of PdM [34]. In this context, there are different types of machine learning (ML) techniques that have been used in various phases of PdM implementation [19], [38]. Recent review studies performed by Lee *et al.* [33] and Kim *et al.* [29], acknowledge that ML approaches provide increasingly effective solutions in these different phases of PdM, facilitated by the growing capabilities of hardware, cloud-based solutions, and newly introduced state-of-the-art algorithms based on different learning methods such as: supervised, unsupervised, semi-supervised, and reinforcement learning [3].

Cluster analysis [27] is one of the most commonly used unsupervised ML technique within exploratory data analysis for identification of hidden patterns. By identifying the cluster structure in the data and augmenting such structure with domain knowledge and meta data, new information can be gained which can further be used in a supervised setting or anomaly detection for the purpose of early detection and prediction of breakdown. In terms of machine knowledge discovery, Diaz-Rozo *et al.* [17] proposed an unsupervised approach for identifying multidimensional patterns of components within a cyber-physical system (CPS) application. In order to find the behavior patterns without any contextual information like machine status or maintenance history, different clustering algorithms such as k-means, hierarchical agglomerative and GMM were applied by using real data acquired through an embedded electronic based CPS device from a computer numeric control (CNC) machine during high throughput machining operation. Then, these algorithms were compared with each other in terms of their contribution to the knowledge extraction about the component performance. As a result, it has been demonstrated that unsupervised ML algorithms embedded in CPS are the key enablers for working towards highly accurate diagnosis and prognosis tools. In another study proposed by the same authors [18], they presented a new unsupervised learning algorithm based on GMM called Gaussian-based dynamic

probabilistic clustering that is capable of coping with data streams and process dynamics. The algorithm was tested by using synthetic data and streams from an industrial test bed. Amruthnath and Gupta [3] proposed an unsupervised learning based approach for early fault detection in PdM. Both GMM and k-means was tested by using real world vibration data collected from an exhaust fan and it was found that GMM is more capable of predicting fault states than k-means. Yuan *et al.* [55] proposed a new unsupervised approach to overcome challenges in feature extraction and segmentation of high dimensional complex condition-based maintenance (CBM) life cycle data. Within such context, they developed two kinds of autoencoder models based on deep learning and cluster analysis and tested them by using an experimental bearing life cycle data set used for fault diagnosis in PdM. Recently, Zschech *et al.* [58] proposed a data science based approach including several analytical methods such as agglomerative hierarchical clustering for unsupervised pattern detection and a recurrent neural network for prognostic model training. The developed approach was tested using real world data in order to demonstrate its usability and practical applicability in the context of building prognostic decision models with missing labels. According to these studies, cluster analysis provides high flexibility and effectiveness in PdM applications for handling common challenges in real world machine and process data such as high dimensionality and lack of correct labeling of machine status or maintenance history [16]. However, the domain is still quite challenging as it requires rather complex analysis of multiple data streams and their interrelation together with a model that enables for interpretable and actionable patterns that can be used as a decision support system for PdM in manufacturing [30].

Related to our point of foci, i.e., analysis of multiple machines, there is a term ''fleet based PHM'' where a fleet is identified as a set of machines grouped with regards to some characteristics, e.g., performing similar tasks, having similar service times or similar performance and health conditions [35], [52]. Fleet based approaches have several advantages. First, it increases the amount of data available for health assessment and remaining useful life calculation [52]. Second, based on similarities of machines, conditions and situations within the fleet allow transfer of knowledge which further improve algorithmic capabilities [51].

Third, this type of approach allows analysis in dynamic operational conditions and detection of deviations on machine behavior by comparisons, with the other machines within the fleet, and offers high level of interpretability as demonstrated in a recent study by Hendrickx *et al.* [23]. Several approaches for PHM of fleets have recently been proposed in the literature [39]. These approaches are usually designed for specific fleet compositions and characteristics [52]. For instance, Lapira [31] proposed a framework for health monitoring and diagnosis of a fleet of wind turbines using clustering. Similarly, Liu [37] proposed a CPS framework for health management of wind turbines and demonstrated the health assessment using fleet-based analytics. Peysson *et al.* [41] has recently presented a framework for fleet-wide maintenance.

Within our paper, we prefer the use of the term ''multi-machine analysis'', rather than ''fleet'' since the latter term inevitable leads one to think more large scale when it comes to the number of machines whilst our framework can provide significant insights with as little as two machines. In terms of selecting the machines taking part in the analysis, we agree with the characteristic that Wagner *et al.* [52] and Lee *et al.* [35] list. As a further specification with respect to our framework, the critical component within the machines should have comparable physical properties and functions since the same cluster patterns are shared over the machines.

In contrast to these previous studies, we deploy a topic model [8], [53] specifically designed to handle group data with shared patterns. Moreover, we integrate this model with an extension to groups of a previous researched framework [10] that are able to handle multiple streams. The result is a coherent framework for multi-machine analysis using multiple data streams and further contributes to the research gap on how to efficiently model a set of machine characteristics in terms of patterns over time in a variety of operation conditions.

III. MULTI-MACHINE ANALYSIS

We first present an overview of our framework for multi-machine analysis. Our main assumption is that we have a set of machines that share some characteristics so that it would be meaningful to have a common language in terms of patterns across those machines. This would allow one to explore the patterns and correlate them with the machines' historical maintenance records, as well as other meta information in order to understand their meaning in terms of degradation [31, Section 2.2.3], as shown in Figure 1.

The main technique used for this purpose is unsupervised learning through cluster analysis which can be defined as grouping collection of data points based on some pattern similarity [27]. In order to obtain such cluster patterns over multiple streams within a machine, we first define a set of features and perform a sliding window operation to extract those features over all streams to obtain a joint feature vector [10], as shown in Figure 2. This allows us to describe the current state of the machine for each point in time corresponding to the sliding window, as shown in Figure 3. This approach is repeated for all machines which results in one set of data points from all machines. We apply a group clustering model [53] commonly referred to as topic modeling in the case of text mining [8]. In this hierarchical model, the clusters are shared at global level and the data residing within groups, i.e., the different machines, can only be modeled through these clusters although with different proportions. Hence, certain machines might have a high percentage of a certain cluster while another has a low percentage, as shown in Figure 4. Now, since the data points are extracted using a sliding window, we can color time segments of the streams of each machine by the cluster identity for that specific period

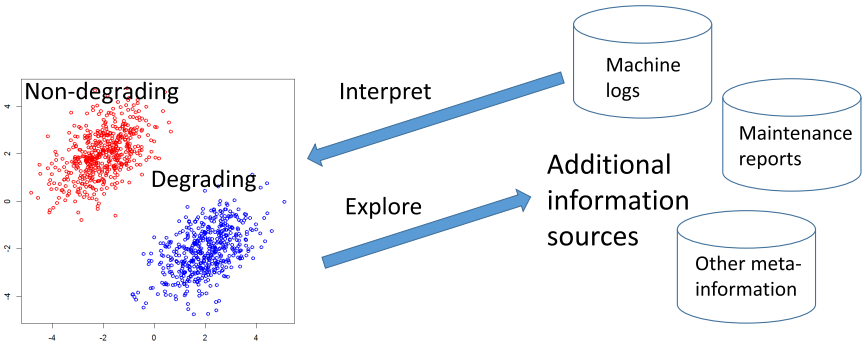


FIGURE 1. An illustration of the framework where clusters are utilized for exploration of machine behavior.

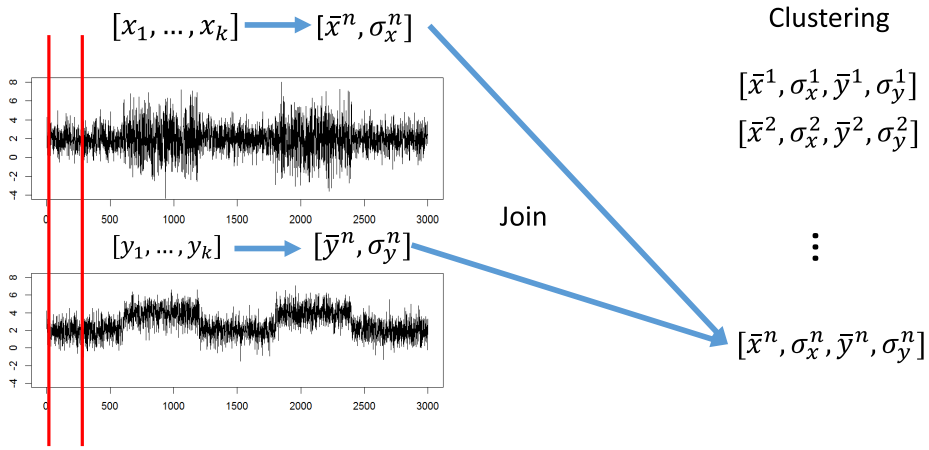


FIGURE 2. An illustration of the data extraction process of the framework.

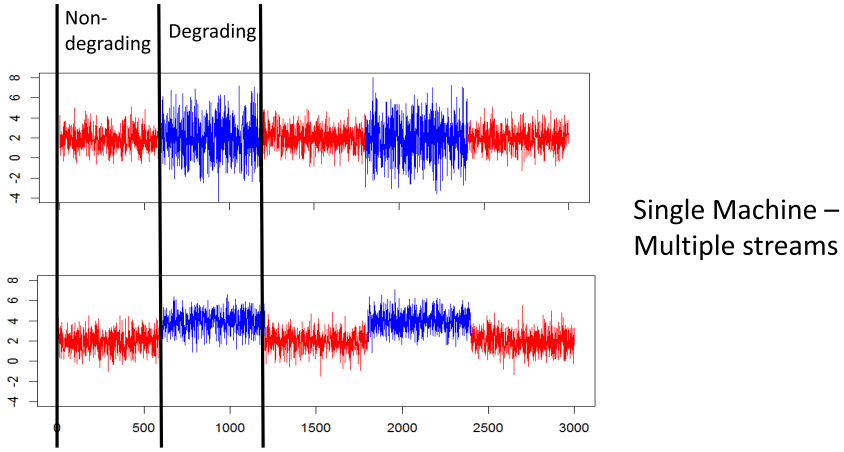


FIGURE 3. An illustration of clusters projected on the streams.

in time [10], [25] in order to be used for visual exploration over time. It should be noted that a given cluster model can be utilized in real-time scenarios for both prediction of class, after labeling the clusters, as well as for anomaly detection, as conceptualized in Figure 5.

IV. ILLUSTRATIVE EXAMPLE

In this section, we provide an illustrative example of the core of our framework by simulated data from multiple machines and streams from those machines. We elaborate on the simulation procedure and the framework including the

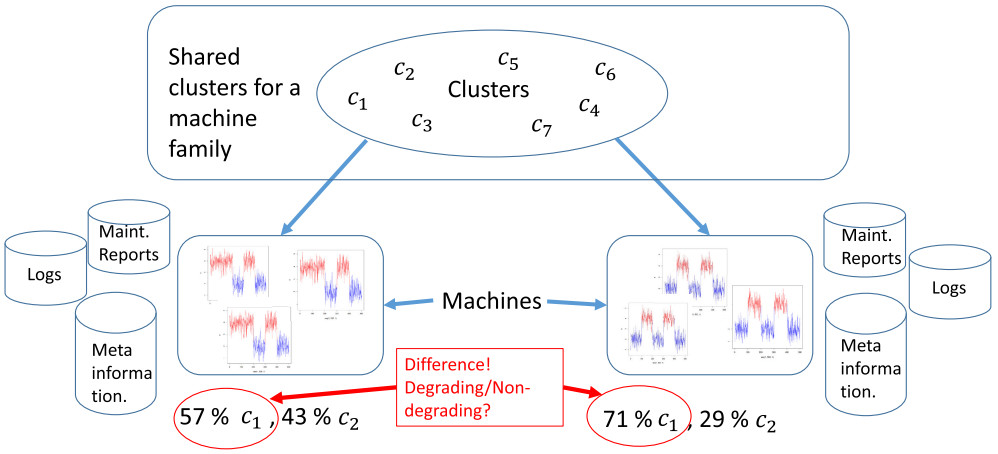


FIGURE 4. An illustration of how shared clusters patterns can be compared across machines.

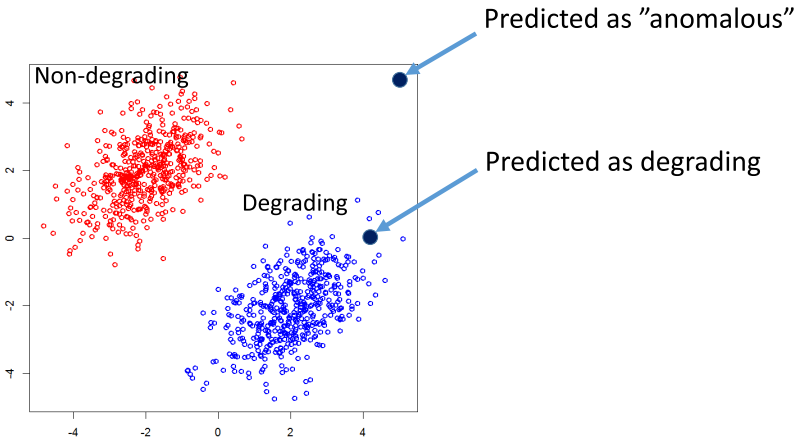


FIGURE 5. A figure that illustrates how a cluster model can be utilized for prediction of the cluster label as well as for anomaly detection.

clustering model. We provide details of the experiment, analyze the results and highlight possible interpretations. Lastly, we perform a comparison with the GMM which is another commonly used cluster model.

A. MULTI-MACHINE SIMULATION

In order to simulate data from multiple streams, we utilize the skewed normal distribution [4] which takes parameters ξ - location, ω – scale, and α – slant.[1] The parameter α regulates the skewness where $\alpha = 0$ results in the standard normal distribution, as shown in Figure 6. In order to simulate data streams, we assume that each stream is defined by certain basic patterns that repeat with some regularity over time. In our case, we also assume that these patterns are defined in terms of different parameter configurations:

$$\vec{\Gamma} \triangleq [\xi, \omega, \alpha]^T, \tag{1}$$

[1]There is an additional parameter τ in the R-package which regulate the specific family of the skewed normal distributions that is being used. We here leave that parameter by its default value, 0, to stay with the skewed normal distribution instead of the extended skewed normal distribution.

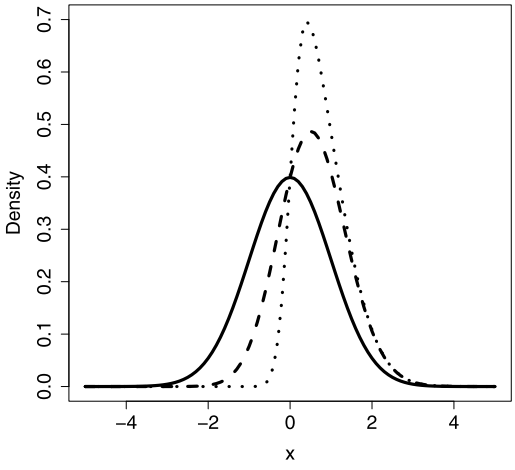


FIGURE 6. The figure depicts the skewed normal distribution for three different levels of slant (skewness): [$\xi = 0, \omega = 1, \alpha = 0$] (solid line); [$\xi = 0, \omega = 1, \alpha = 1$] (dashed line); [$\xi = 0, \omega = 1, \alpha = 4$] (dotted line).

of the skewed normal distribution in order to simulate some deviation from the normal assumption often made and which

our clustering approach is based on (since we use Gaussian mixture components). Moreover, since in our setting, machines consist of m streams, we define a matrix of basic patterns which capture the behavior for each stream in relation to the other streams, i.e., we assume correlation between the streams:

$$A \triangleq [\vec{\Gamma}_1, \ldots, \vec{\Gamma}_m]^T \tag{2}$$

The regularity of patterns is defined in terms of a Markov chain $p(A_i|A_j)$ between different patterns by drawing probability distributions for each stream in the following way:

$$p(A_k|A_l) \sim Dirichlet(\beta_1, \ldots, \beta_h), \tag{3}$$

where $\beta_i = \beta, \forall i \in \{1, \ldots, h\}$, i.e., h basic patterns, and $\beta \leq 1$, i.e., a symmetric "bowl"-shaped Dirichlet density over the patterns emphasizing some stability when transitioning from one pattern to another. Each pattern A_i also has a certain duration time which we simulate utilizing a Poisson distribution $Pois(\lambda)$ by drawing the number of samples for which the pattern is manifested.[2] Lastly, different type of noise may distort the data. In order to reflect it in our simulation model, we implement the concept of ϵ-contamination [5]:

$$\Upsilon([\xi,\ \omega, \alpha]^T, \epsilon) \triangleq (1-\epsilon)\mathcal{N}_s([\xi, \omega, \alpha]^T) + \epsilon\mathcal{N}(\mu, \sigma), \tag{4}$$

where $\mathcal{N}_s$ is the skewed normal density, $\mathcal{N}$ is the normal density and where:

$$\mu \sim Uniform(x; \zeta^{\uparrow}, \zeta^{\downarrow}), \quad \sigma \sim Uniform(x; \eta^{\uparrow}, \eta^{\downarrow}), \tag{5}$$

where $Uniform(\cdot)$ is the uniform distribution over the intervals $[\cdot^{\uparrow}, \cdot^{\downarrow}]$. The idea behind this type of contamination is to have some distortion noise according to a degree specified by ϵ (a small continuous number).

Now, given the set of h basic patterns for the streams:

$$\mathcal{A} \triangleq \{A_1, \ldots, A_h\}, \tag{6}$$

we can construct the following simulation configuration, to be used in Algorithm 1, for obtaining multiple streams from one machine:

$$[\mathcal{A}, \beta, \lambda, \epsilon, \zeta^{\uparrow}, \zeta^{\downarrow}, \eta^{\uparrow}, \eta^{\downarrow}]^T, \tag{7}$$

Since we can generate data streams for a single machine, we utilize this algorithm in order to generate a set of machines, all of which are sharing patterns to different extent, as depicted by 2.

B. GAUSSIAN TOPIC MODELING

By utilizing the algorithms in the previous section, we assume that we have obtained streams of sensor measurements $x^1_{1:n}, \ldots, x^m_{1:n}$, where $x^i_{1:n} \triangleq x^i_1, \ldots, x^i_n$, from each machine. We then proceed by extracting features over a sample point window of size w and extract u features $f_{1:u}(\vec{x}) \triangleq f_1(\vec{x}), \ldots, f_u(\vec{x})$ over that window for each stream, yielding joint aggregated feature vectors (in block matrix notation) of the form [10], [25]:

$$\vec{F}_k \triangleq \left[f_{1:u}\left(\left[x^1_{k:(k+w-1)}\right]^T\right), \ldots, f_{1:u}\left(\left[x^m_{k:(k+w-1)}\right]^T\right)\right]^T \tag{8}$$

where k is the starting index for the window. Then, by letting the window move over time and extract features, sets of data vectors are generated. However, if one move the window with only one step, many similar joint feature points may be generated since then only one point within the window has changed. In practice, one can therefore choose to let the window "jump" a number of steps r [25], which yields points of the form:

$$\mathcal{F} \triangleq \left[\vec{F}_1, \vec{F}_{r+1}, \vec{F}_{2r+1}, \ldots, \vec{F}_{\lfloor \frac{n-w}{r} \rfloor r+1}\right] \tag{9}$$

and since our simulation procedure, as defined by Algorithm 2, ensures that each machine $\mathcal{M}_\bullet$ has the same number of streams and samples within those streams, we can obtain a collection/groups of data points, $[\mathcal{F}_{1:v}]^T$ where v is the number of machines under consideration. Given this collection of data, we apply a GTM [8], [53], where each topic is a multivariate Gaussian distribution. For clarity, we utilize the term cluster instead of topic in the description below. The generative structure of the model (for details, see further Section IV-C) can then be described by:

1) Draw q clusters (topics) in terms of covariance matrices $\Sigma_{1:q}$, and means $\vec{\mu}_{1:q}$ based on some appropriate prior (further specified later in Section IV-C).
2) Draw group cluster proportions $\vec{\theta}_{1:v}$ (also further specified in Section IV-C).
3) For each machine $i \in \{1, \ldots, v\}$:
   a) Draw a cluster $c \sim Categorical(\vec{\theta}_i)$
   b) Draw a vector $\vec{F} \sim Normal(\vec{\mu}_c, \Sigma_c)$
   c) Repeat from 3a

The key feature of this type of hierarchical model is that it captures the most important patterns on a shared global level. The aggregated data for each machine can only be modeled through these shared clusters and, hence, one obtain a common terminology in terms of patterns across machines which can be further used for exploration and interpretation using some additional meta-information such as logs of machines and historical maintenance reports. Combining such information can yield significant insights in degrading performance and breakdown behavior. We utilize the multivariate Gaussian distribution as a representation for clusters due to the fact that it is a well-known and an easy interpretable distribution which has been found useful in clustering problems within manufacturing [3], [17]. It should be noted that we implicitly make the assumption that clusters have multi-dimensional convex elliptic shape by using multivariate Gaussian distributions as cluster components. However, as we will highlight in the next section, we will utilize probabilistic programming techniques [12] where the model is specified through a programming language separated from the training algorithm which

[2] We will assume the same sampling frequency for all streams.

Algorithm 1: Simulate Single Machine – Multiple Streams

Input : Simulation parameters $[\mathcal{A}, \beta, \lambda, \epsilon, \zeta^{\uparrow}, \zeta^{\downarrow}, \eta^{\uparrow}, \eta^{\downarrow}]^T$, a transition matrix $p(A_k|A_l)$ and a number of samples n

Output: m data streams consisting of n points denoted by $x^1_{1:n}, \ldots, x^m_{1:n}$

$i \leftarrow 0$

Draw initial pattern $A_i \sim Uniform(\mathcal{A})$

repeat

  Draw pattern $A_{i+1} \sim p(A_k|A_i)$, $k \in \{1, \ldots, h\}$, see Equation (3)

  Draw duration of A_{i+1} in terms of number of samples $z_{i+1} \sim Pois(\lambda)$

  for $j \leftarrow 1$ **to** m **do**

    sample $x^j_{\left(\sum_{k=1}^{i-1} z_k+1\right):\left(\sum_{k=1}^{i-1} z_k+z_i\right)} \sim \Upsilon\left(A_{i+1}\langle j, \rangle^T, \epsilon\right)$, where $A_{i+1}\langle j, \rangle$ is the $j:th$ row of A_{i+1}

  end

  $i \leftarrow i + 1$

until $\sum_{k=1}^{i-1} z_k \geq n$

For each stream $x^1_{1:n}, \ldots, x^m_{1:n}$ remove points with index above n (spill over)

return $x^1_{1:n}, \ldots, x^m_{1:n}$

Algorithm 2: Simulate Multiple Machines - Multiple Streams

input : Simulation parameters $[\mathcal{A}, \beta, \lambda, \epsilon, \zeta^{\uparrow}, \zeta^{\downarrow}, \eta^{\uparrow}, \eta^{\downarrow}]^T$, and a number of samples n

output: Collections of data $\mathcal{D}_1, \ldots, \mathcal{D}_v$ corresponding to machines $1, \ldots, v$

Draw v transition matrices $p_{1:v}$ using Equation (3)

for $i \leftarrow 1$ **to** v **do**

  call Algorithm 1 with transition matrix p_i and gather data into collection $\mathcal{D}_i$

end

return $\mathcal{D}_1, \ldots, \mathcal{D}_v$

makes it possible to easily change the cluster/component distribution according to the problem at hand.

C. EXPERIMENT DESIGN

The simulation procedure is instantiated by defining the configuration parameters according to:

$$\mathcal{A} \triangleq \left\{ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 2 & 1 & 2 \\ 2 & 2 & 4 \\ 2 & 1 & -4 \end{bmatrix}, \begin{bmatrix} -1 & 3 & -2 \\ -1 & 2 & -2 \\ -1 & 1 & -2 \end{bmatrix} \right\} \tag{10}$$

$$[\beta, \lambda, \epsilon] \triangleq [0.1, 50, 0.01] \tag{11}$$

$$[\zeta^{\uparrow}, \zeta^{\downarrow}] \triangleq [-1, 1] \tag{12}$$

$$[\eta^{\uparrow}, \eta^{\downarrow}] \triangleq [1, 2]. \tag{13}$$

Then, we select five machines to simulate by using Algorithm 2, each of them contains three data streams with 1000 samples per data stream. For simplicity, we choose to extract two basic features, namely the mean and standard deviation [25], over a window of size $w = 10$, and a jump between windows of $r = 5$. The resulting streams for each machine is depicted in Figure 7.

In order to implement[3] the GTM found in Section IV-B, we utilize probabilistic programming through Stan [12] with the No U-turn sampler (NUTS) for Hamiltonian Monte Carlo (HMC) [24]. We base the GTM code/model on the latent Dirichlet allocation implementation, found in Section 9.5 of the Stan users' guide [47] where the discrete cluster belonging variables have been averaged out and the components in our case are multivariate Gaussian distributions.

There is a general problem of training for mixture models due to non-identifiable and label switching issues [6]. Hence one cannot easily perform posterior analysis without resorting to some additional post processing [21, Section 22.3]. Furthermore, the clustering problem itself is multimodal

[3]Code: http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-19969

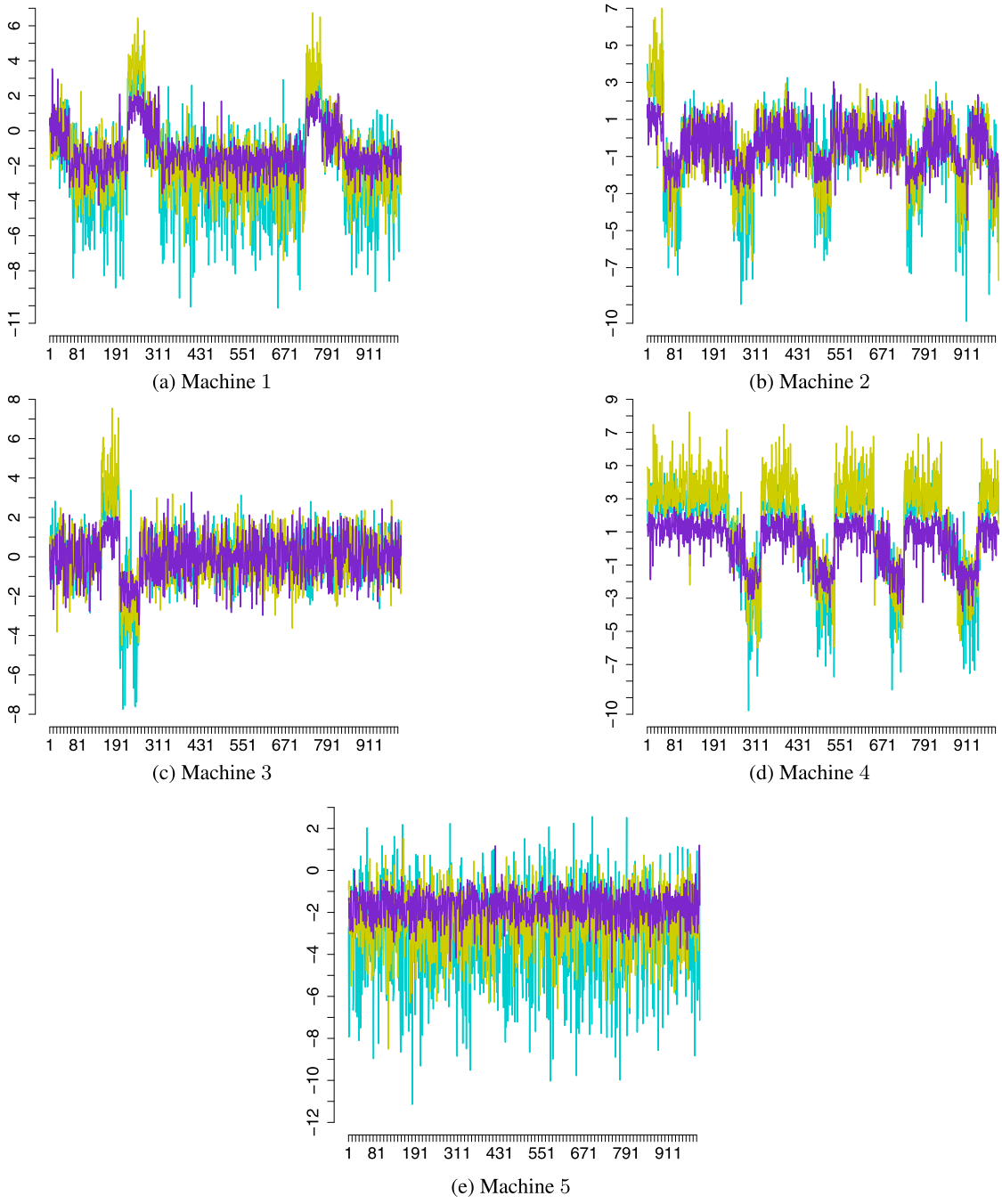
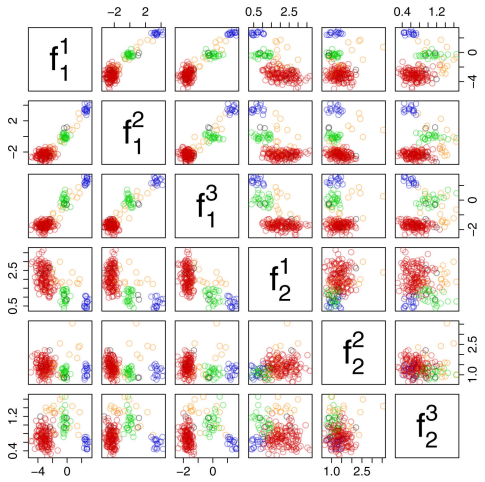


FIGURE 7. Simulated streams for machines $1-5$.

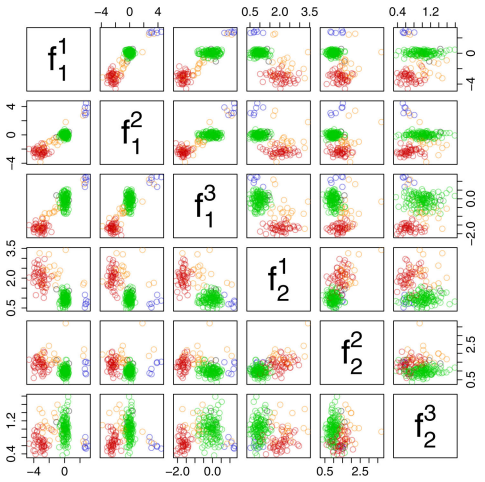
which is a challenge for Markov chain Monte carlo (MCMC) in general since the chain might only explore the surroundings of a single mode dependent on initial conditions and priors. However, for our purpose of exploratory data analysis, we can take a pragmatic view on these issues and therefore, even though only one mode of posterior has been explored, it is acceptable as long as the clustering provides a useful view on the data and is somewhat stable between different runs. This is similar to the implicit view within the topic modeling community as well, since one there often resorts to MCMC by Gibbs sampling [8] which is likely to only explore a single mode [7].

The data is standardized and then, after some experimentation with respect to stability of clustering results, we obtained the following priors (see further Section IV-B):

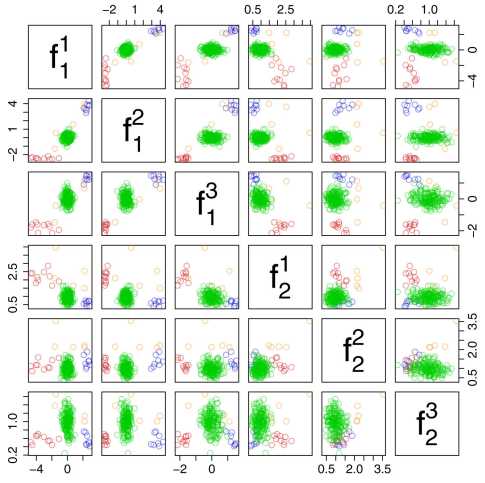
$$\vec{\mu}_i \langle l \rangle \sim Normal(0, 1), \quad l \in \{1, \ldots, u \times m\}$$
$$\Sigma_i \triangleq \text{diag}(\vec{\kappa}_i)\ \Omega\ \text{diag}(\vec{\kappa}_i)$$



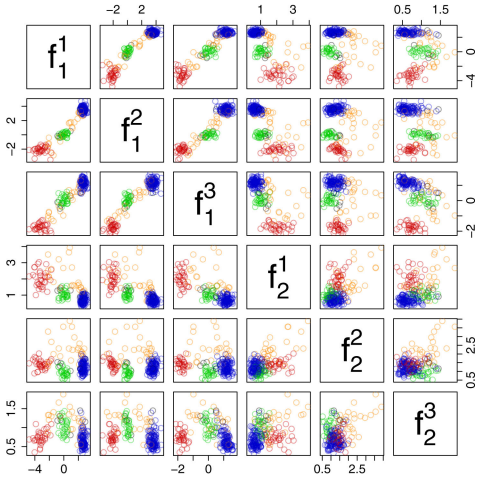
(a) Machine 1



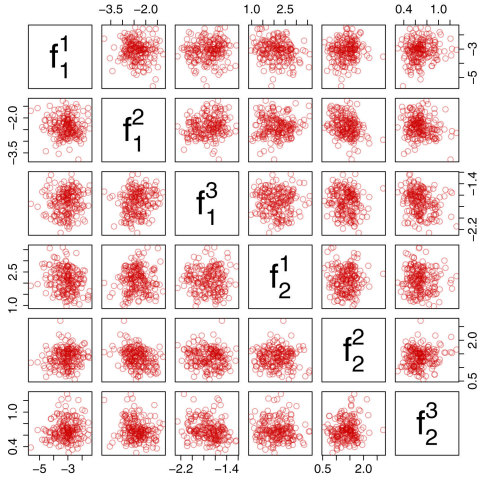
(b) Machine 2



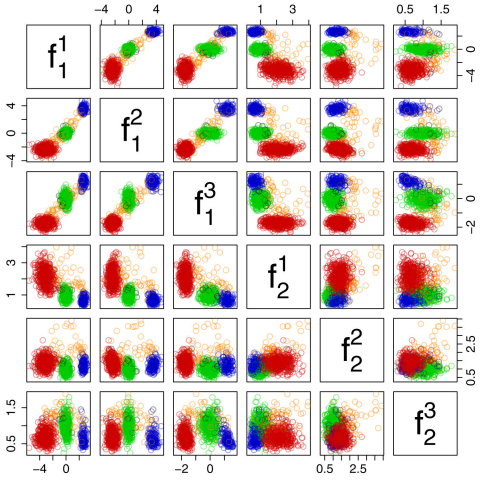
(c) Machine 3



(d) Machine 4



(e) Machine 5



(f) All machines

FIGURE 8. Clusters within machines $1-5$ visualized in pairwise feature space, where f_i^j denotes the feature, and $i \in \{1, 2\}$ correspond to the mean and standard deviation and $j \in \{1, 2, 3\}$ to the streams.

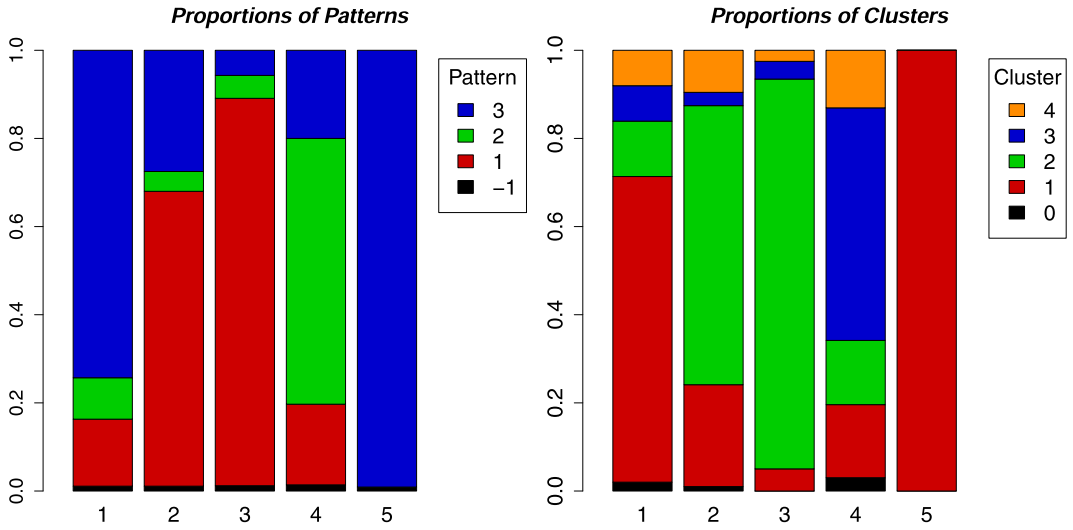


FIGURE 9. Proportions of patterns compared to clusters within machines 1 – 5 as well as all machines together. From the figure, we observe that pattern 1, 2, 3, in Figure 9a, match cluster 2, 3, 1, in Figure 9b.

$$\begin{aligned} \Omega_i &\sim LKJCorr(1) \\ \vec{\kappa}_i\langle l\rangle &\sim Cauchy(1), \quad l \in \{1,\ldots,u\times m\} \\ \vec{\theta}_j &\sim Dirichlet(\beta_1,\ldots,\beta_q), \end{aligned} \tag{14}$$

where $i \in \{1,\ldots,q\}, j \in \{1,\ldots,v\}$, I is the identity matrix, $\vec{\kappa}_i\langle l\rangle$ denote the l : th row of vector $\vec{\kappa}_i$; $\vec{\mu}_{1:q}$, $\Sigma_{1:q}$, $\Omega_{1:q}$ have dimensions in agreement with the number of features and streams, i.e., $u \times m$, corresponding to $\vec{F}$ and $\vec{\theta}$ have a dimension in correspondence to the number of clusters q where $\beta_k = 1, \forall k \in \{1,\ldots,q\}$. LKJCorr is a prior distribution for a correlation matrix, see further [47]. Regarding the standard multivariate normal prior for $\vec{\mu}_{1:q}$, it is rather informative in the sense that the cluster centers should be near the neighborhood of zero in all dimensions, however, more weakly informative priors resulted in less stability in clusters over different runs.

The label switching issue is something that needs to be handled in order to obtain useful clustering results. There are several different methods of different complexities to handle this issue [28], [42], [45], [57]. We take a simple approach for this issue by inspiration from the label allocation approach, see further Zhu and Fan [57], together with a greedy algorithm for the final cluster assignments and leave evaluation of different approaches for future work when applying our method to a real world scenario. Now, for each sample from HMC/NUTS: $\vec{\mu}_{1:q}$, $\Sigma_{1:q}$, $\vec{\theta}_{1:v}$, the probability for a feature vector $\vec{F}_i$ belonging to a specific cluster j has the following appearance:

$$p\big(c_i = j|\vec{\mu}_{1:q}, \Sigma_{1:q}, \vec{\theta}_{1:v}\big) \propto \vec{\theta}_{\omega(i)}\langle j\rangle\, \mathcal{N}(\vec{F}_i|\vec{\mu}_j, \Sigma_j), \tag{15}$$

where $\omega(i)$ and $\omega(j)$ denotes the machine identity of $\vec{F}_i$ and $\vec{F}_j$, respectively; $\vec{\theta}_{\omega(i)}\langle j\rangle$ denotes component j of $\vec{\theta}_{\omega(i)}$ and $\mathcal{N}(\cdot|\cdot)$ is the normal density function. Given this we can proceed by formulating Algorithm 3 which simply assigns the component (cluster) with the maximum probability to the data point.[4] By running this algorithm on all samples: $\vec{\mu}^i_{1:q}$, $\Sigma^i_{1:q}$, $\vec{\theta}^i_{1:v}$, $i \in \{1,\ldots,s\}$ obtained from HMC/NUTS, we get s different clusterings which we gather in a $s \times t$ cluster matrix C where t is the number of points to label. This matrix will then be the input to Algorithm 4 which will create a final clustering given a certain threshold δ that determines the level of acceptable uncertainty regarding the clustering. Such technique is inspired by previous research [50] where multiple labels have been assigned to cluster points but where we here omit such a multi-assignment and instead declare points that do not reach the threshold limit as uncertain with respect to cluster identity. The algorithm proceeds by first counting the number of times any two given points have ended up in the same cluster and then, given a random permutation, cluster points will be assigned to the same cluster if two points have at least appeared in the same cluster over the samples with a probability at least as high as δ. Points that are not assigned to any clusters (in the algorithm assigned to -1) will then be treated as uncertain with regards to cluster belonging. Lastly, as a final step, the clusters will be ordered according to size and then the q largest of these will be assigned to a final cluster label while the remaining points will be assigned to -1 as an indication that there exists some uncertainty regarding those points.

In general, there is an issue in selecting the number of clusters, however, in this illustrative example, for simplicity, we use a simple procedure by a pairs plot where one can see nuances of three clusters and some intermediate points that needs to be modeled by some auxiliary cluster, yielding a total of four clusters in the model, i.e., $q = 4$, and we also set the threshold to $\delta = 0.7$ and save further exploration of the impact of this threshold to future research.

[4]This algorithm as well as Algorithm 4 have more efficient implementations but we omit such details here for clarity.

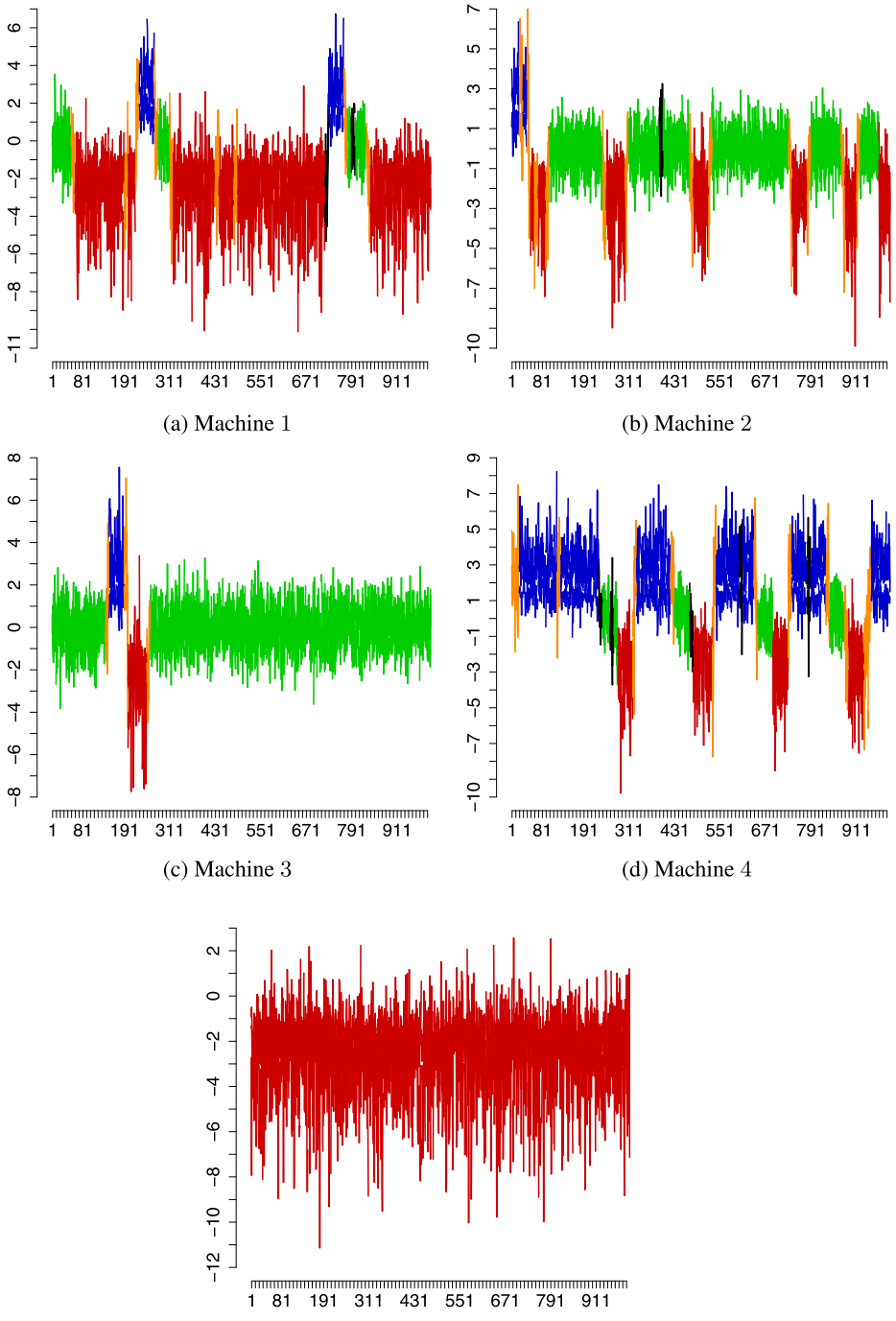


FIGURE 10. **Streams for machines 1 – 5 colored by cluster identity.**

D. RESULTS

The resulting clusters for each individual machine as well as for all machines together can be seen in Figure 8. We observe that for some machines, a majority of the points have been assigned to a single cluster while others to a larger extent, contain two or more clusters. Due to the nature of data preprocessing by a sliding window, there also exists an intermediate cluster in between the main clusters. The relation between the proportion of the patterns in the data, as defined by $\mathcal{A}$ in Equation (10), and the proportions of identified clusters by Algorithm 3 and 4 are shown in Figure 9. We observe that the GTM model / algorithms well identify the proportion of patterns that are evident in each machine (the orange cluster is the intermediate cluster which can in in principle be ignored).

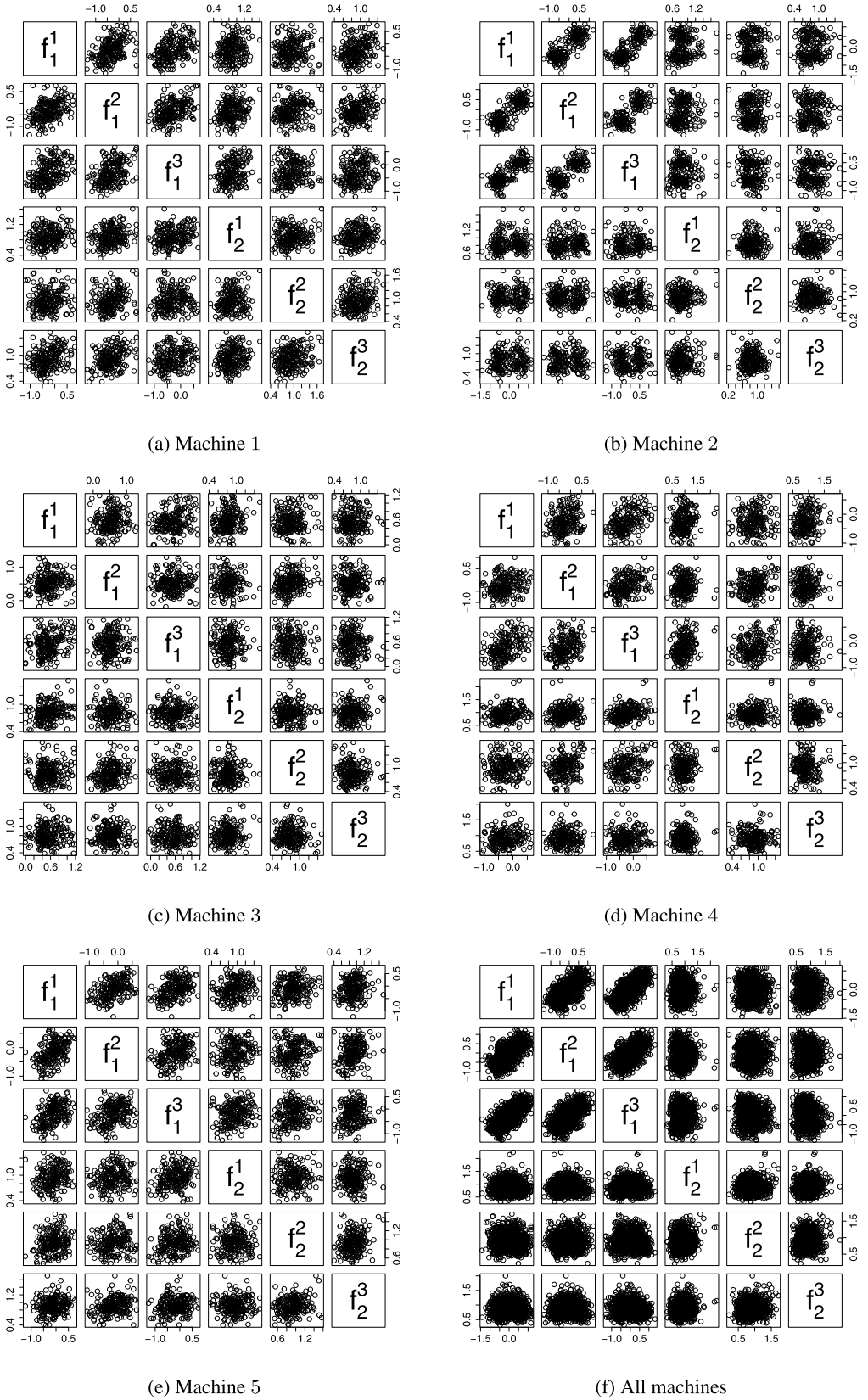
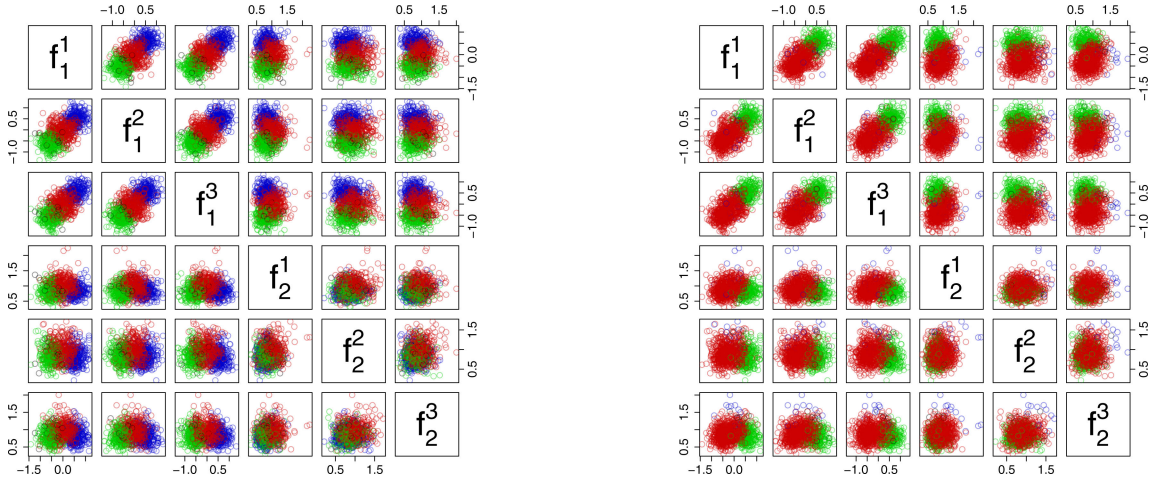


FIGURE 11. Data for scenario as described by Equation (16) within machines $1-5$ visualized in pairwise feature space, where f_i^j denotes the feature, and $i \in 1, 2$ corresponds to the mean and standard deviation and $j \in \{1, 2, 3\}$ to the streams.



(a) Clustering results of GTM

(b) Clustering results of GMM

FIGURE 12. **Results of GMM and GTM clustering on the scenario defined by Equation** (16).

Since we also utilize a sliding window, we can color each sliding window part of the streams by their cluster belongings as illustrated in Figure 10 [10], [25]. However, due to the size of the windows $w = 10$ and we have data points for each jump of $r = 5$, the windows are overlapping which means that individual points might belong to two different clusters. We here choose to color the streams by window indices: $1, 1 + w, 1 + 2w$, etc. corresponding to mutually exclusive windows.

E. EXPLORATION AND INTERPRETATION

Running the GTM over aggregated streams within multiple machines is the first step of the exploration phase. By using additional information such as logs, maintenance reports and meta-information one can correlate patterns within the machines with their degradation behaviors. Moreover, by using multi-machine analysis, there can be a possibility to compare gained information from a certain machine pattern found in the reports with the identified patterns across several machines. In this context, the identified cluster patterns becomes a unified language for the set of machines which enables one to project ''degradation hypotheses'' from one machine to another.

As an example, considering Figure 10, if one can identify a certain degradation behavior in the reports for machine 3, one can hypothesize that this behavior is related to the green cluster pattern and hence, from that perspective one can further analyze the degradation behavior of primarily machine 2, since a large proportion of that cluster appears there, then secondly one can further analyze machine 4 and 1, since those machines also contain some small proportion of the cluster. This is a good example where one single machine guides further analysis towards certain other machines with similar behavior in terms of cluster patterns. There might also be cluster patterns that jointly are the cause for degradation behavior. For instance if machine 4 is the main concern when it comes to degradation, then the continuous interchange between the blue, green and red clusters would be the main subject for further exploration.

Depending on the application scenario, the clusters can be interpreted in different ways. As an example in multi-tooling machines, the cluster patterns can capture different modes of operation, e.g., in terms of vibration. In older machines, where the current operation mode cannot be extracted easily from the control units of the machines, this presents a way of obtaining such information and subsequently allows for analysis of the machines' operation mode. In other cases, the cluster patterns can capture the different variations that might occur to different extent within a certain mode of operation and degradation state. Based on such clusters and previous knowledge of machine degradation, one can identify future system degradation states, e.g., healthy, warning, and faulty. Capturing those cluster patterns and explore to what extent they appear in different machines with different degradation states then becomes the point of foci.

F. COMPARISON WITH GAUSSIAN MIXTURE MODEL

Since the GTM is designed to find clusters on a group level, one can obtain cluster information which might not be so easily captured by other clustering methods that do not take individual groups into account. The reason for this is that certain clusters might appear more clearly on certain machines which then helps the model to identify these clusters. This can be illustrated by another simulation scenario, using the same pre-processing as before, where the clusters are not equally well separated, defined by Equation 16.

$$\mathcal{A}' \triangleq \left\{ \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & -1 \\ 0 & 1 & -1 \\ 0 & 1 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \right\} \tag{16}$$

Algorithm 3: Single Sample Clustering

Input : A sample $\vec{\mu}_{1:q}$, $\Sigma_{1:q}$, $\vec{\theta}_{1:v}$, a $t \times (u \times m)$ matrix $[\mathcal{F}_{1:v}]^T$ (block format) and rows $\vec{F}_i^T \triangleq [\mathcal{F}_{1:v}]^T \langle i, \rangle$

Output: Cluster labels $c_{1:t} \in \{1, \ldots, q\}$, corresponding to $\vec{F}_{1:t}^T$

```
Declare label vector c_i, ∀i ∈ {1, ..., t}
Declare temporary unnormalized probability p_u^temp
Declare maximum unnormalized probability p_u^max
for i ← 1 to t do
    p_u^max ← 0
    for j ← 1 to q do
        p_u^temp ← θ_ω(i) ⟨j⟩ N(F_i | μ_j, Σ_j) as defined in Equation (15)
        if p_u^temp > p_u^max then                    ▷ Update cluster belonging
            c_i ← j
            p_u^max ← p_u^temp
        end
    end
end
return c_i ∈ {1, ..., q}, i ∈ {1, ..., t}             ▷ return cluster assignments
```

An example of simulated data from this scenario where the clustering results differ between the GTM and the GMM is shown in Figure 11. From Figure 11b, nuances of clusters appear since one can observe some separations between groups of points. This is information that will be beneficial for the GTM during training since it has a group level. On the other hand, the GMM, which is not taking individual groups into account, will be trained with a view on data from all machines, shown in Figure 11f, where the previously mentioned nuances of clusters are not at all evident. A simple experiment on this data, running the GTM and GMM 30 times with the same parameters as priors set according to Equation (14), where the GMM implementation is based on the previously described GTM implementation but without the group level and where the number of clusters is set to 3 and threshold set to $\delta = 0.7$, results in that the GTM captures the clusters, with some variations, in all of the cases, while the GMM captures all the clusters in only 8 of the cases. The result of the first run of the experiment is shown in Figure 12. As it is seen, the red and green cluster in Figure 12a are consistent with the previously mentioned nuances of clusters from Figure 11b and 11d. In 22 cases, the GMM only correctly identified one of the cluster, shown in green in Figure 12b, i.e., it failed to separate the actual two clusters that resides within the red cluster area in Figure 12b. This is due to that the model is performing clustering on all data points without the group level, as shown in Figure 11f, where the clusters are not as clearly separated. It should also be noted that other choices of priors can yield different results that might help identifying the cluster for the case of GMM also, however, the point we want to make here, given that the priors are exactly the same for the GMM and GTM in this case, is that the hierarchical structure, taking groups into account, is an advantage of the GTM which could decide whether or not certain clusters are identified.

To summarize, due to the fact that the GMM does not model group information and hence does not consider or strive at being a good fit to data within individual groups; it can lose information for identifying clusters and can therefore lose track of possibly key cluster patterns. It should be noted that training a cluster model with obtained stability between clustering results is a difficult task due to the multimodality of such problems. Hence, different initialization can yield convergence to different modes unless there are indications of clusters as is the case on the group level in our example.

V. DISCUSSION

By using a framework where cluster patterns are identified across machines, such as we propose, one increases the likelihood of taking full benefits of any limited fault event data associated to any of the involved machines. As an example, if certain patterns appear to be rather unique for a subset of machines which have had several faulty conditions, one may use the model to observe any tendency of such patterns in the other machines and act accordingly. In such cases, any contextual information, e.g., in terms of more in-depth knowledge of behavior for certain machines, can also play a crucial role for understanding how the identified patterns can be projected and used for the other machines.

Algorithm 4: δ-Clustering

Input : A $s \times t$ clustering matrix $C \triangleq [[c^1_{1:t}]^T, \ldots, [c^s_{1:t}]^T]^T$ and a probability thresholds δ

Output: A clustering $\tilde{c}_{1:t}$, with labels in $\{1, \ldots, q\}$

Initialize $count[i, j] \leftarrow 0$ for $i, j \in \{1, \ldots, t\}$ ▷ number of times points are in the same cluster
Initialize $\tilde{c}_i \leftarrow 0, \forall i \in \{1, \ldots, t\}$ ▷ 0 means not yet assigned
for $i \leftarrow 1$ **to** s **do** ▷ start counting
  for $j \leftarrow 1$ **to** t **do**
    for $k \leftarrow 1$ **to** t **do**
      if $C\langle i, j\rangle = C\langle i, k\rangle$ **then**
        $count[j, k] \leftarrow count[j, k] + 1$
      end
    end
  end
end
Draw random permutation $\pi : \{1, \ldots, t\} \mapsto \{1, \ldots, t\}$
$curr \leftarrow 1$
for $i \leftarrow 1$ **to** t **do** ▷ iterate and assign all points above the threshold
  if $\tilde{c}_{\pi(i)} = 0$ **then**
    $\tilde{c}_{\pi(i)} \leftarrow curr$
    for $j \leftarrow 1$ **to** t **do**
      if $\tilde{c}_j = 0$ **then**
        $p \leftarrow count[\pi(i), j]/s$
        if $p \geq \delta$ **then**
          $\tilde{c}_j \leftarrow curr$ ▷ assign cluster label
        end
      end
    end
    $curr \leftarrow curr + 1$
  end
end
Construct cluster partitions $\mathcal{C}_j \triangleq \{i : \tilde{c}_i = j, i \in \{1, \ldots, t\}\}$, where $j \in \{\tilde{c}_{1:t}\}$
Construct ordering $\Psi : \{\tilde{c}_{1:t}\} \mapsto \{\tilde{c}_{1:t}\}$ so that $\left|\mathcal{C}_{\Psi(i)}\right| \geq \left|\mathcal{C}_{\Psi(j)}\right|$ iff $i < j$ ▷ decreasing order
Relabel $\tilde{c}_i \leftarrow j$ for $i \in \mathcal{C}_{\Psi(j)}$ where $j \in \{1, \ldots, q\}$ ▷ the q largest clusters will be returned
Relabel $\tilde{c}_i \leftarrow 0$ for $i \in \mathcal{C}_{\Psi(j)}$ where $j \in \{q + 1, \ldots, |\{\tilde{c}_{1:t}\}|\}$ ▷ uncertain (unassigned) points
return $\tilde{c}_{1:t}$ ▷ return a q-clustering

The parameters involved in the framework, in particular window and jump size as well as the number of clusters, need to be adjusted to the problem characteristic at hand. As an example, if degradation is expected to appear over several weeks one may aggregate data over days, i.e. a window size corresponding to a day and set the jump size to non-overlapping windows. A smaller jump size can be also set in order to obtain overlapping windows which will increase the smoothness of extracted features. This can be beneficial for visual exploration of cluster results. However, in case of massive data, the window and jump size can be used as a way of reducing the number of extracted feature

data points. This can be particular useful in our case where MCMC is used since such approaches are computationally demanding [11]. Also for larger data sets, the δ-algorithm may need further adaptions in order to decrease computational complexity. For adjustment of parameters as well as different additions of extracted features, partly determining the dimension, we propose that one uses visual exploration to learn about the effect on the results both in terms of patterns captured as well as with respect to interpretation of those patterns.

Besides being able to capture different patterns over multiple streams and machines, the framework can also be utilized to capture normal operations of machines and be used as an anomaly detector [15], [53]. In contrast to clustering models that do not account for the group level information, the framework can also capture anomalies in terms of pattern proportions, i.e., not only deviations from the patterns but also how much of those patterns within machines change. This might reveal significant insights into degradation behavior.

However, as previously noted [18], degradation over time is likely to smoothly change previously identified patterns. Therefore, another important part of the clustering problem in an online setting is to let the model self-learn and self-adapt to normal tolerable changes over time without raising an anomaly alarm. This is commonly referred to as concept drift [20] in the machine learning community, i.e., the process itself is not stable over time which means that learned models trained on data in the beginning and end of the process are different to some extent. By using Bayesian theory [22], this can be handled by setting the posterior to the prior and retrain with a certain period [1]. By doing this, the model drifts ''along'' with the degradation behavior, i.e., the changes of patterns over time, without raising unnecessary alarms. These changes of the model might also provide in itself valuable information in terms of machine degradation behavior, since one can then capture how much the model has changed over time and store this information as historical records of drifting behavior. For example, if a certain pattern is highly evident in the start of the analysis, but then at some point starts to change over time, one might correlate such information with other meta information regarding the machine and learn the cause of such behavior.

Lastly, more advanced models, which also have the ability to self-adapt to the number of clusters [48], [49], can allow one to explore machine behavior in terms of splitting and merging of cluster patterns over time, denoted as evolutionary clustering [14], [54], [56]. Such splitting and merging behavior might further be matched with other meta information in order to understand why they occur and what they mean in terms of degradation behavior.

VI. CONCLUSION

We have presented a coherent framework which is able to capture information, at different levels, from the repetitive structure that multiple machines constitutes. The framework is based on a hierarchical probabilistic model, denoted GTM [53], where shared cluster patterns are captured at the top level and then used at different proportions for modeling the data, within the machines, on a group level. In order to obtain single data points over time from each machine, we aggregate a number features over all the data streams for a specific time window [10], [25]. The framework uses a clustering schema, through Algorithm 1 and 4, which takes uncertainty into account and where a threshold regulates a desirable degree of reliability of assignment to clusters. We have illustrated our framework through synthetic data by simulating multiple machines/streams and where we performed modeling through probabilistic programming in Stan [12]. We showed that the model can capture different patterns and proportions of those patterns well across several machines and we highlighted potentially utilization as information gain through association with additional meta information and comparison of the patterns across machines. Moreover, we also performed an experiment that highlights the benefit of the GTM through capturing the group level information to identify clusters. In contrast, the commonly used GMM operates on all data, i.e., does not take individual groups into account, and hence only aims to be a good fit of all data without the group perspective. As it has been seen from the experiment, this can result in the GMM is not able to identify clusters at the group level, and can thereby miss capturing key patterns in the data which might be crucial for the understanding of machine's degradation behavior. As a conclusion, it is important to highlight that such a framework contributes to fill the gap in data driven based PdM application in real word industrial environments since it deals with the challenge of lack of labeled data coming from multiple streams across multiple machines.

For our future work, we will apply our framework on real-world manufacturing case studies. One issue that one needs to handle is the big data aspect [43]. Currently, statistical machine learning in the form of Bayesian inference [22] do not enjoy the same scalability as for instance deep learning methods [32] where it is usually easy to parallelize. However, recently proposed methods [11] suggest that there are opportunities to scale down the data size significantly and still obtain useful results. Such methods might provide the ideal complement to probabilistic programming tools such as Stan [47] and would then enable for a fast modeling and training loop, i.e., ''small data'' for efficient and effective exploratory multi-machine analysis.

REFERENCES

[1] L. AlSumait, D. Barbará, and C. Domeniconi, ''On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking,'' in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 3–12.

[2] P. Daugherty, P. Banerjee, W. Negm, and A. E. Alter, ''Driving unconventional growth through the industrial Internet of Things,'' Accenture Technol., Tech. Rep., 2015.

[3] N. Amruthnath and T. Gupta, ''Fault class prediction in unsupervised learning using model-based clustering approach,'' in *Proc. Int. Conf. Inf. Comput. Technol. (ICICT)*, Mar. 2018, pp. 5–12.

[4] A. Azzalini. (2020). The R Package SN: The Skew-Normal and Related Distributions Such as the Skew-*t* (Version 1.6-2). Università di Padova, Italia. [Online]. Available: http://azzalini.stat.unipd.it/SN

[5] J. Berger and L. M. Berliner, "Robust Bayes and empirical Bayes analysis with ϵ-contaminated priors," *Ann. Statist.*, vol. 14, no. 2, pp. 461–486, 1986.

[6] M. Betancourt. (2017). *Identifying Bayesian Mixture Models*. Accessed: Oct. 31, 2019. [Online]. Available: https://mc-stan.org/users/documentation/case-studies/identifying_mixture_models.html

[7] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, 2017.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[9] J. Bokrantz, A. Skoogh, C. Berlin, T. Wuest, and J. Stahre, "Smart maintenance: A research agenda for industrial maintenance management," *Int. J. Prod. Econ.*, vol. 224, Jun. 2020, Art. no. 107547.

[10] M.-R. Bouguelia, A. Karlsson, S. Pashami, S. Nowaczyk, and A. Holst, "Mode tracking using multiple data streams," *Inf. Fusion*, vol. 43, pp. 33–46, Sep. 2018.

[11] T. Campbell and T. Broderick, "Automated scalable Bayesian inference via Hilbert coresets," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 551–588, 2019.

[12] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language," *J. Stat. Softw.*, vol. 76, no. 1, pp. 1–32, 2017.

[13] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. D. P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Comput. Ind. Eng.*, vol. 137, Nov. 2019, Art. no. 106024.

[14] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2006, pp. 554–560.

[15] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.

[16] S. Cho, G. May, I. Tourkogiorgis, R. Perez, O. Lazaro, B. D. L. Maza, and D. Kiritsis, "A hybrid machine learning approach for predictive maintenance in smart factories of the future," in *Proc. IFIP Int. Conf. Adv. Prod. Manage. Syst.* Cham, Switzerland: Springer, 2018, pp. 311–317.

[17] J. Diaz-Rozo, C. Bielza, and P. Larrañaga, "Machine learning-based CPS for clustering high throughput machining cycle conditions," *Procedia Manuf.*, vol. 10, pp. 997–1008, Jan. 2017.

[18] J. Diaz-Rozo, C. Bielza, and P. Larranaga, "Clustering of data streams with dynamic Gaussian mixture models: An IoT application in industrial processes," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3533–3547, Oct. 2018.

[19] M. Fernandes, A. Canito, V. Bolón-Canedo, L. Conceição, I. Praça, and G. Marreiros, "Data analysis and feature selection for predictive maintenance: A case-study in the metallurgic industry," *Int. J. Inf. Manage.*, vol. 46, pp. 252–262, Jun. 2019.

[20] J. Gama, I. Žliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, p. 44, 2014.

[21] A. Gelman, B. J. Carlin, S. H. Stern, B. D. Dunson, A. Vehtari, and D. Rubin, *Bayesian Data Analysis*. Boca Raton, FL, USA: CRC Press, 2004.

[22] A. Gelman, B. J. Carlin, S. H. Stern, and B. D. Rubin, *Bayesian Data Analysis*. Boca Raton, FL, USA: CRC Press, 2004.

[23] K. Hendrickx, W. Meert, Y. Mollet, J. Gyselinck, B. Cornelis, K. Gryllias, and J. Davis, "A general anomaly detection framework for fleet-based condition monitoring of machines," *Mech. Syst. Signal Process.*, vol. 139, May 2020, Art. no. 106585.

[24] M. D. Homan and A. Gelman, "The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1593–1623, 2014.

[25] A. Holst, J. Bae, A. Karlsson, and M.-R. Bouguelia, "Interactive clustering for exploring multiple data streams at different time scales and granularity," in *Proc. Workshop Interact. Data Mining*, 2019, p. 2.

[26] B. Hoppenstedt, R. Pryss, B. Stelzer, F. Meyer-Brötz, K. Kammerer, A. Treß, and M. Reichert, "Techniques and emerging trends for state of the art equipment maintenance systems—A bibliometric analysis," *Appl. Sci.*, vol. 8, no. 6, p. 916, Jun. 2018.

[27] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[28] A. Jasra, C. C. Holmes, and D. A. Stephens, "Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling," *Stat. Sci.*, vol. 20, no. 1, pp. 50–67, Feb. 2005.

[29] D.-H. Kim, T. J. Y. Kim, X. Wang, M. Kim, Y.-J. Quan, J. W. Oh, S.-H. Min, H. Kim, B. Bhandari, I. Yang, and S.-H. Ahn, "Smart machining process using machine learning: A review and perspective on machining industry," *Int. J. Precis. Eng. Manuf.-Green Technol.*, vol. 5, no. 4, pp. 555–568, Aug. 2018.

[30] P. Klein and R. Bergmann, "Generation of complex data for AI-based predictive maintenance research with a physical factory model," in *Proc. 16th Int. Conf. Informat. Control, Autom. Robot. (ICINCO)*, Prague, Czech Republic, 2019, pp. 40–50.

[31] E. R. Lapira, "Fault detection in a network of similar machines using clustering approach," M.S. thesis, Univ. Cincinnati, Cincinnati, OH, USA, 2012.

[32] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[33] G.-Y. Lee, M. Kim, Y.-J. Quan, M.-S. Kim, T. J. Y. Kim, H.-S. Yoon, S. Min, D.-H. Kim, J.-W. Mun, J. W. Oh, I. G. Choi, C.-S. Kim, W.-S. Chu, J. Yang, B. Bhandari, C.-M. Lee, J.-B. Ihn, and S.-H. Ahn, "Machine health management in smart factory: A review," *J. Mech. Sci. Technol.*, vol. 32, no. 3, pp. 987–1009, 2018.

[34] J. Lee, C. Jin, Z. Liu, and H. D. Ardakani, "Introduction to data-driven methodologies for prognostics and health management," in *Probabilistic Prognostics and Health Management of Energy Systems*. Cham, Switzerland: Springer, 2017, pp. 9–32.

[35] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications," *Mech. Syst. Signal Process.*, vol. 42, nos. 1–2, pp. 314–334, Jan. 2014.

[36] U. Leturiondo, O. Salgado, L. Ciani, D. Galar, and M. Catelani, "Architecture for hybrid modelling and its application to diagnosis and prognosis with missing data," *Measurement*, vol. 108, pp. 152–162, Oct. 2017.

[37] Z. Liu, "Cyber-physical system augmented prognostics and health management for fleet-based systems," Ph.D. dissertation, Dept. Mech. Mater. Eng., Univ. Cincinnati, Cincinnati, OH, USA, 2018.

[38] E. Lughofer and M. Sayed-Mouchaweh, *Predictive Maintenance in Dynamic Systems: Advanced Methods, Decision Support Tools and Real-World Applications*. Cham, Switzerland: Springer, 2019.

[39] G. Michau, T. Palmé, and O. Fink, "Fleet PHM for critical systems: Bi-level deep learning approach for fault detection," in *Proc. Eur. Conf. PHM Soc.*, vol. 4, 2018, p. 403.

[40] K. T. P. Nguyen and K. Medjaher, "A new dynamic predictive maintenance framework using deep learning for failure prognostics," *Rel. Eng. Syst. Saf.*, vol. 188, pp. 251–262, Aug. 2019.

[41] F. Peysson, C. Mozzati, D. Leon, Q. Lafuste, and J.-B. Leger, "Fleet-wide proactive maintenance of machine tools," in *Twin-Control*. Cham, Switzerland: Springer, 2019, pp. 209–224.

[42] S. Richardson and P. J. Green, "On Bayesian analysis of mixtures with an unknown number of components (with discussion)," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 59, no. 4, pp. 731–792, 1997.

[43] S. Sagiroglu and D. Sinanc, "Big data: A review," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2013, pp. 42–47.

[44] S. Selcuk, "Predictive maintenance, its implementation and latest trends," *Proc. Inst. Mech. Eng. B, J. Eng. Manuf.*, vol. 231, no. 9, pp. 1670–1679, Jul. 2017.

[45] M. Stephens, "Dealing with label switching in mixture models," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 62, no. 4, pp. 795–809, 2000.

[46] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 812–820, Jun. 2015.

[47] *Stan User's Guide*, Version 2.21, Stan Develop. Team. [Online]. Available: https://mc-stan.org/about/

[48] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Sharing clusters among related groups: Hierarchical Dirichlet processes," in *Advances in Neural Information Processing Systems*, vol. 17, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA, USA: MIT Press, 2005, pp. 1385–1392.

[49] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *J. Amer. Statist. Assoc.*, vol. 101, no. 476, pp. 1566–1581, 2006.

[50] B. Ulfenborg, A. Karlsson, M. Riveiro, C. X. Andersson, P. Sartipy, and J. Synnergren, "Multi-assignment clustering: Machine learning from a biological perspective," *J. Biotechnol.*, vol. 326, pp. 1–10, Jan. 2021.

[51] A. Voisin, G. Medina-Oliva, M. Monnin, J.-B. Leger, and B. Iung, "Fleet-wide diagnostic and prognostic assessment," in *Proc. Annu. Conf. Prognostics Health Manage. Soc.*, 2013, pp. 1–10.
[52] C. Wagner, P. Saalmann, and B. Hellingrath, "An overview of useful data and analyzing techniques for improved multivariate diagnostics and prognostics in condition-based maintenance," in *Proc. Annu. Conf. Prognostics Health Manage. Soc.*, 2016, pp. 3–6.
[53] L. Xiong, B. Póczos, J. Schneider, A. Connolly, and J. VanderPlas, "Hierarchical probabilistic models for group anomaly detection," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 789–797.
[54] T. Xu, Z. Zhang, P. S. Yu, and B. Long, "Dirichlet process based evolutionary clustering," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 648–657.
[55] J. Yuan, K. Wang, and Y. Wang, "Deep learning approach to multiple features sequence analysis in predictive maintenance," in *Proc. Int. Workshop Adv. Manuf. Autom.* Singapore: Springer, 2017, pp. 581–590.
[56] J. Zhang, Y. Song, C. Zhang, and S. Liu, "Evolutionary hierarchical Dirichlet processes for multiple correlated time-varying corpora," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 1079–1088.
[57] W. Zhu and Y. Fan, "Relabelling algorithms for mixture models with applications for large data sets," *J. Stat. Comput. Simul.*, vol. 86, no. 2, pp. 394–413, Jan. 2016.
[58] P. Zschech, K. Heinrich, R. Bink, and J. S. Neufeld, "Prognostic model development with missing labels," *Bus. Inf. Syst. Eng.*, vol. 61, no. 3, pp. 327–343, Jun. 2019.

ALEXANDER KARLSSON received the Ph.D. degree in computer science from Örebro University, in 2010. He is currently a member of the research group Skövde Artificial Intelligence Laboratory, University of Skövde, Sweden. His research interest includes probabilistic modeling for information- and data science using Bayesian theory. His main research interest is within data science are exploratory data analysis for the purpose of decision-making. He has made a number of contributions to the research area of information fusion and has long term experience in collaboration with Swedish industry. His current research interest includes within the area of customized statistical data analysis and obtaining fast modeling loops for the purpose of knowledge extraction. He received the title of Docent at the University of Skövde, in 2019.

EBRU TURANOGLU BEKAR received the Ph.D. degree in industrial engineering from Dokuz Eylöl University, Turkey, in December 2016. She is currently a Postdoctoral Researcher with the Production Systems Division, Department of Industrial and Materials Science, Chalmers University of Technology. She is also a Research Group Member of Production Service and Maintenance Systems. She has performed research activities with a number of contributions in areas, such as total productive maintenance, performance measurement of manufacturing systems using artificial intelligence, multi-criteria decision-making, engineering statistics, and quality control. The goal of her current research is to build a structured way to analyze industrial big data and develop algorithms based on artificial intelligence/machine learning techniques in smart maintenance.

ANDERS SKOOGH is currently a Professor of production maintenance with the Department of Industrial and Materials Science, Chalmers University of Technology. He is also a Research Group Leader for Production Service and Maintenance Systems. He is also the Director of Chalmers' master's program in production engineering and a Board Member of Sustainability Circle (www.sustainabilitycircle.se) with responsibilities for research collaboration.