Collaborative Prognostics for Machine Fleets Using a Novel Federated Baseline Learner

Vibhor Pandhare[1], Xiaodong Jia[1], and Jay Lee[1]

[1]*University of Cincinnati, Cincinnati, OH, 45221, USA*

pandhavr@mail.uc.edu
jiaxg@ucmail.uc.edu
lj2@ucmail.uc.edu

ABSTRACT

Difficulty in obtaining enough run-to-fail datasets is a major barrier that impedes the widespread acceptance of Prognostic and Health Management (PHM) technology in many applications. Recent progress in federated learning demonstrates great potential to overcome such difficulty because it allows one to train PHM models based on distributed databases without direct data sharing. Therefore, this technology can overcome local data scarcity challenges by training the PHM model based on multi-party databases. To demonstrate the ability of federated learning to enhance the robustness and reliability of PHM models, this paper proposes a novel federated Gaussian Mixture Model (GMM) algorithm to build universal baseline models based on distributed databases. A systematic methodology to perform collaborative prognostics is further presented using the proposed federated GMM algorithm. The usefulness and performance are validated through a simulated dataset and the NASA Turbofan Engine Dataset. The proposed federated approach with parameter sharing is shown to perform at par with the traditional approach with data sharing. The proposed model further demonstrates improved robustness of predictions made collaboratively keeping the data private compared to local predictions. Federated collaborative learning can serve as a catalyst for the adaptation of business models based on the servitization of assets in the era of Industry 4.0. The methodology facilitates effective learning of asset health conditions for data-scarce organizations by collaborating with other organizations preserving data privacy. This is most suitable for a servitization model for Overall Equipment Manufacturers who sell to multiple organizations.

1. INTRODUCTION

The advent of technology paradigms such as Artificial Intelligence (AI), Internet of Things (IoT), Cyber-physical Systems (CPS), and initiatives like Industry 4.0 direct to a future where intelligence is endowed to every value-adding entity across enterprises. Pervasive sensor networks with cheaper and convenient computational capabilities (Lee, Davari, Singh & Pandhare, 2018) form the foundational infrastructure for these paradigms. Building on this foundation, the real fruits of sustainable and synergistic profit can only be reaped through collaboration between these entities. One of the most significant outcomes of collaborative operation is servitization (Palau, Dhada, Bakliwal & Parlikad, 2019; Baines, Lightfoot, Evans, Neely, Greenough, Peppard, Roy, Shehab, Braganza, Tiwari & Alcock, 2007): selling of services instead of selling the assets that provide the service. In the servitization model, the client does not pay for owning the asset but pays for their right to use the asset. In such a scenario, upkeep of the assets is even more important as downtime directly transfers to the revenue lost. But developing a prognostics model can be challenging in practice for multiple reasons. Despite the availability of sensors, the frequency of “event” data per asset that represents an equipment failure can be low due to the high reliability of machines. Additionally, due to operational variation, one machine cannot present all degradation patterns in a lifetime. Preparing comprehensive datasets within a single enterprise is highly impractical given the range of assets and operational conditions per enterprise. Attempting to develop models from experimentation is not just costly and time-consuming but can also lack consistency with field behavior. Thus, collaboration among industrial entities for developing reliable and robust prognosis models can unlock huge potential for improved maintenance decisions.

However, traditional machine learning approaches necessitate the accumulation of data at a location, such as data centers or cloud-servers, for training analytical models. There are two major challenges associated with this. Primarily, with the volume, velocity, and variety of the data generation, transferring of data is not efficient. Advancements in cloud-based solutions only address the issue of centralized storage through developing architectures such as edge computing and fog computing (Yi, Li, & Li, 2015). Secondly, the transfer of data can violate data-ownership laws as well as raise privacy concerns for collaborating entities. This is especially true for Overall Equipment Manufacturers (OEM) who sell to multiple clients. OEMs may not be able to harvest the data while the equipment is in use at the client site, as the data is considered proprietary. McKinsey & Company reports data ownership is a major barrier for manufacturing companies toward collaborating, even with third-party providers such as scientists, etc. (McKinsey, 2016). Beyond OEMs, governments across the world are strengthening laws to protect data privacy (Regulation (EU), 2016). This issue of data privacy becomes an irony as it blocks the very foundational path of collaboration to Industry 4.0's success. Thus, new learning schemes need to be developed that can unlock the potential of distributed private data for collaborative prognostics.

Multiple methods have come up in recent times to address specific problems that come along with the increasing ubiquity of data and learning from it. Batch learning is developed to solve the problem of inefficient model training due to computation memory exhaustion while training on large data sets (Bisong, 2019). Transfer Learning is developed to avoid needing large datasets and retraining models from scratch to learn similar and related patterns (Pan & Yang, 2009). More recently, Federated learning, proposed by Google, is the radical solution to overcome the bottleneck of learning from datasets that are distributed over multiple systems keeping the local datasets private (Yang, Liu, Chen & Tong, 2019). The novelty lies in training a global machine learning model only via communication of parameters and parameter updates between each system. While most of the research in federated learning has been focused on applications about users (Yang, Liu, Chen & Tong, 2019), such as text prediction using Natural Language Processing, making recommendations for retail by estimating user profiles, etc., industrial applications remain largely untapped. As Industrial Artificial Intelligence (AI) possesses its own set of challenges compared to general-purpose AI, similarly Federated learning also brings its challenges for adaptation to industrial problems. Thus, a novel methodology for performing collaborative prognostics is proposed, which includes a new algorithm called Federated Expectation Maximization for privacy-preserving model aggregation.

The remainder of the article is organized as follows. Section 2 presents a literature review. Section 3 introduces the proposed federated Expectation Maximization algorithm for training multivariate Gaussian Mixture Model (GMM). Section 4 illustrates the application of the proposed algorithm using two case studies for parameter estimation and collaborative fault prognosis. Section 5 concludes the article.

2. LITERATURE REVIEW

Collaborative learning has recently gained significant attention, although being considered for a long time under various self-organizational architectures such as Multi-agent Systems and Holonic Systems. In the area of Prognostics and Health Management (PHM), Palau, et.al. (2019) propose a multi-agent system for real-time distributed collaborative prognostics. They show that distributed collaborative prognostics are advantageous in scenarios with significant fleet heterogeneity, limited computing capability, and faulty measurements. Information is shared between assets assigned to groups formed using a *Friendship Matrix*, and predictions are made using Weibull-Time-To-Event Recurrent Neural Networks (WTTE-RNN) trained on run-to-failure trajectories. Additionally, Lin, Liu, Byon, Qian, Liu & Huang (2017) propose a collaborative learning framework for estimating many individualized regression models in a heterogeneous population of run-to-failure trajectories. However, depending on the degradation trajectory of a system it may not always be optimal to capture the behavior in a single failure model (Lei, Li, Guo, Li, Yan & Lin, 2018). Thus, the prediction is often preceded by fault detection or health-stage separation. Fault detection involves assessing the health condition of equipment to determine whether an event can be considered as an occurrence of fault (Lapira, 2012). It is difficult and unnecessary to predict the remaining useful life in the healthy stage as it contains no information about the degradation trend. Thus, fault detection becomes a critical primary step to perform effective prognostics. Most works on collaborative fault detection approach the problem from a similarity-based clustering or peer-to-peer comparison methodology for a fleet of assets. Zhao, Li, Lu, Lv, Gu & Shang (2020) implemented a fault detection model using collaborative filtering techniques for detecting an incipient fault in large-scale solar farms by sharing current data among photovoltaic systems. Maroli, Özgüner & Redmill (2019) propose a collaborative fault detection framework for large-scale vehicle networks using an echo state network. Ng & Srinivasan (2010) developed a multi-agent-based collaborative fault detection and identification system for application in chemical process plants and combines various heterogeneous methods to maximize performance using information fusion. These approaches exploit data sharing facilities to derive inferences based on the assumption that the fleet belongs within a single plant or an organization with data-privacy not being a concern. For collaborative learning across organizations where data sharing is not possible, privacy-preserving learning frameworks such as federated learning can be implemented.

The development of approaches for federated learning has been largely neural network-based. With the availability of data significantly increasing in recent times, deep learning has been used extensively for wide-ranging applications including for PHM (Rosero, Silva & Ribeiro, 2020). Neural Networks provide a convenient option for training over a federated setting. Weights and biases of the global network can be initiated in the parameter server which can be then communicated with the clients. At each client node, the gradient can be evaluated from the local dataset based on the present parameters of the network and the gradient can be shared back to the server where it can be aggregated using algorithms like federated averaging (Konečný, McMahan, Yu, Richtárik, Suresh & Bacon, 2016; Dhada, Parlikad & Salvador, 2020). Over multiple rounds of communication, the global model can be trained to keep the data private. Numerous upgradations have been performed in this architecture to address issues such as gradient quantization (Jin, Huang, He, Dai & Wu, 2020), reducing expensive communications (Li, Sahu, Talwalkar, & Smith, 2020), addressing system heterogeneity, efficient gradient aggregation, and especially security concerns through homomorphic encryption, differential privacy, etc. However, almost all of the methods discussed in the federated learning literature are developed for the task of supervised learning in data abundant scenarios, i.e. they assume the availability of large datasets with corresponding labels, which is not always possible. While statistical machine learning methods provide the high capability to capture the knowledge from limited data using an unsupervised approach, they remain largely unexplored for federated implementations. Thus, there is a lack of systematic analytical methodology to build prognostic models based on decentralized databases addressing limitations of data-privacy as well as local label unavailability. A systematic federated approach is proposed to facilitate collaborative prognostics.

3. PROPOSED METHOD

Mixture models, especially Gaussian Mixture Model (GMM), are a widely used statistical method as an effective universal approximator. Justifiably, it finds use in several applications (Pimentel, Clifton, Clifton & Tarassenko, 2014; Diaz-Rozo, Bielza & Larrañaga, 2020) such as density estimation, clustering, association rules, outlier detection, latent factors, ranking, and even data visualization. Given its wide use, effective training of GMM is a continuously evolving area (Jin, Zhang, Balakrishnan, Wainwright & Jordan, 2016; Kurban, Jenne, & Dalkilic, 2017) with Expectation Maximization (EM) being one of the popular methods (Ververidis & Kotropoulos, 2008; Balakrishnan, Wainwright & Yu, 2017; Zhao, Li & Sun, 2020). Thus, federated expectation maximization for the training of GMM is proposed to build a universal baseline model in the problem of collaborative prognostics. This forms the first novelty of the work. Additionally, a systematic methodology to perform collaborative prognostics based on the proposed algorithm is also presented. This forms the second novelty of the work. The improvement in performance and robustness using the proposed approach is validated through the NASA Turbofan Engine Dataset (Saxena & Goebel, 2008).

3.1. Gaussian Mixture Model and EM Algorithm

Given data D=$\{x_i\}_{i=1,\dots,M}$, the GMM has the form

$$p(x_i|\mathbf{\Theta}) = \sum_{k=1}^{K} c_k \cdot N(x_i|\mu_k, \Sigma_k) \tag{1}$$

where Kdenotes the number of components or clusters. $\mathbf{\Theta} = \{c_k, \mu_k, \Sigma_k\}_{k=1,\dots,K}$ are the unknown parameters to be estimated by Expectation Minimization (EM) algorithms, which includes the mixing weights $c_k \in [0,1]$ and $\sum_{k=1}^{K} c_k = 1$, the mean μ_k and the covariance Σ_k for the k-th component. Membership of point x_i to cluster k are described as the posterior probability

$$r_{i,k} \triangleq p(z_i = k|x_i, \mathbf{\Theta}) \tag{2}$$

where $z_i \in \{1, \dots, K\}$ is the latent state and $\mathbf{z} = \{z_i\}_{i=1,\dots,M}$ forms the latent vector.

Parameters $\mathbf{z}$ and $\mathbf{\Theta}$ in GMM is estimated by Expectation Maximization (EM) algorithm, which maximizes the lower bound of Component Data Log Likelihood (CDLL) through the alternation between the E-step and M-step. CDLL has the form

$$CDLL(\mathbf{\Theta}) = \sum_{i=1}^{M} \log p(x_i, z_i|\mathbf{\Theta}) \tag{3}$$

CDLL in (3) is difficult to compute since z_i is unknown. Therefore, EM optimizes the Expected Complete data Log Likelihood (ECLL) instead, which describes the lower bound of CDLL. The E-step and M-Step in EM algorithm can be written as:

```
Loop until converge
{
E-step:   $ECLL(\mathbf{\Theta}, \mathbf{\Theta}^{(t-1)})$
          $= \mathbf{E}_z[CDLL(\mathbf{\Theta})|\mathbf{\Theta}, \mathbf{\Theta}^{(t-1)}]$          (4)
M-step:   $\mathbf{\Theta}^{(t)} = \operatorname*{argmax}_{\mathbf{\Theta}} ECLL(\mathbf{\Theta}, \mathbf{\Theta}^{(t-1)})$   (5)
}
```

where $\mathbf{\Theta}^{(t-1)}$ is the estimated $\mathbf{\Theta}$ from the previous step. Based on the discussions, the algorithm for the parameter estimation of GMM can be detailed as given in Fig. 1.

3.2. Federated GMM for universal baseline modeling

The proposed method builds the GMM model from decentralized databases as shown in Fig. 2. In the setting of machine PHM, the local client servers in Fig. 2 represent the database at different organizations, manufacturing plants, corporations, etc. Normally, cross-organization data-sharing

Algorithm 1

```
Initialize $\boldsymbol{\Theta}$ as $\boldsymbol{\Theta}_0$ and assign $\boldsymbol{\Theta}^{(t-1)} = \boldsymbol{\Theta}_0$
1)  Perform E-step and compute the membership
      $r_{ik} = \frac{c_k^{(t-1)} \cdot p\left(x_i | \mu_k^{(t-1)}, \Sigma_k^{(t-1)}\right)}{\sum_{k'} c_{k'}^{(t-1)} \cdot p\left(x_i | \mu_{k'}^{(t-1)}, \Sigma_{k'}^{(t-1)}\right)}$
2)  Perform M-step and update the $\boldsymbol{\Theta}^{(t-1)}$ to $\boldsymbol{\Theta}^{(t)}$
      $c_k^{(t)} = \frac{1}{M} \sum_i r_{i,k}$
      $\mu_k^{(t)} = \frac{\sum_i r_{ik} \cdot x_i}{c_k^{(t)}}$
      $\Sigma_k^{(t)} = \frac{\sum_i r_{ik} \cdot \left(x_i - \mu_k^{(t)}\right)\left(x_i - \mu_k^{(t)}\right)^T}{c_k^{(t)}}$
3)  Repeat 1) and 2) until converge.
```

Fig. 1 Algorithm for standard GMM

Algorithm 2

```
Give data $\mathcal{D}=\{\mathcal{D}_u\}_{u=1,\dots,U}$,where $\mathcal{D}_u = \{x_i^u\}_{i=1,\dots,M_u}$ is the data from local client $u$.
Initialize $\boldsymbol{\Theta}$ as $\boldsymbol{\Theta}_0$
assign $\boldsymbol{\Theta}^{(t-1)} = \boldsymbol{\Theta}_0 = \left\{c_k^{(t-1)}, \mu_k^{(t-1)}, \Sigma_k^{(t-1)}\right\}_{k=1,\dots,K}$
where $K$ is the number of components in GMM
1)  For each **client** $u$, compute the membership
      $r_{ik}^u = \frac{c_k^{(t-1)} \cdot p\left(x_i^u | \mu_k^{(t-1)}, \Sigma_k^{(t-1)}\right)}{\sum_{k'} c_{k'}^{(t-1)} \cdot p\left(x_i^u | \mu_{k'}^{(t-1)}, \Sigma_{k'}^{(t-1)}\right)}$
    and the local summation term:
      $SX_k^u = \sum_{i=1}^{M_u} r_{ik}^u \cdot x_i^u$ and $SC_k^u = \sum_{i=1}^{M_u} r_{ik}^u$
    Send $\{SX_k^u, SC_k^u\}_{k=1,..,K}$ to the global server
2)  At the global **server**, compute the updated mixing weights and the global mean.
      $c_k^{(t)} = \frac{1}{\sum_u M_u} \cdot \sum_u SC_k^u$ ;     $\mu_k^{(t)} = \frac{\sum_u SX_k^u}{c_k^{(t)}}$
    Send global mean $\left\{\mu_k^{(t)}\right\}_{k=1,\dots,K}$ to each client server

3)  For each **client** $u$, compute the local covariance
      $SV_k^u = \sum_{i=1}^{M_u} r_{ik}^u \cdot \left(x_i^u - \mu_k^{(t)}\right)\left(x_i^u - \mu_k^{(t)}\right)^T$
    Send $\{SV_k^u\}_{k=1,..,K}$ to the global server

4)  At the global **server**, compute the updated covariance
      $\Sigma_k^{(t)} = \frac{\sum_u SV_k^u}{c_k^{(t)}}$;
    Send global covariance $\left\{\Sigma_k^{(t)}\right\}_{k=1,\dots,K}$ to each client server
5)  Repeat 1) ~ 4) until the stopping criteria is met
```

Fig. 3 Algorithm Description for Federated GMM

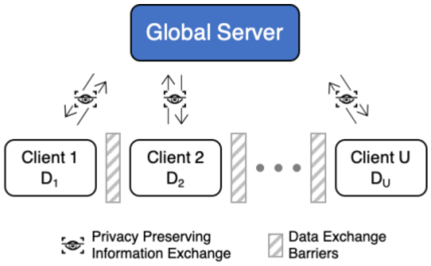

Fig. 2 Decentralized databases for model training

is not favored due to security concerns, administrative reasons, and many others. To overcome the barrier of data-sharing, this study proposes a novel algorithm called Federated Expectation Maximization to develop a Gaussian Mixture Model without sharing data across organizations. The execution steps of the proposed algorithm are outlined in Fig 3. In a broader impact, the proposed modifications to the GMM can be easily extended to any other EM-based algorithms and most MLE-based algorithms for parameter estimation. Based on the information flow outlined in Algorithm 2, secure data exchange between the global server and local clients can be promoted by homomorphic encryption, which will allow computation on the encrypted text directly at the global server. Adding homomorphic encryption to the proposed algorithm will only affect the computation efficiency. The discussions about homomorphic encryption in left for future work. The goal of this study is to demonstrate and validate the performance of the proposed algorithm only.

3.3. Proposed methodology for failure prognosis

The proposed methodology is presented in Fig. 4. The training set consists of clients (or units) with run-to-failure data, and the testing set can be units with any amount of data available. The data for each unit can be distributed across organizations, plants, etc. in such a way that data-sharing is not allowed. Federated training of the global baseline GMM model in the global server is performed by the exchange of model parameters with each client using the local baseline data as described in the proposed algorithm in Fig. 3. The deviation metric representing the health of the unit using the trained gaussian mixture model is given by weighted squared Mahalanobis Distance (MD). The deviation metric is evaluated for each client locally. MD is defined by (6), where $\boldsymbol{x}_m^u$ denote the sensor measurements for a sample, and $\boldsymbol{\Theta} = \{c_k, \mu_k, \Sigma_k\}_{k=1,\dots,K}$ denote the parameters of the trained global baseline model.

$$
\begin{aligned}
r &= d_u(\boldsymbol{x}_m^u \mid \boldsymbol{\Theta}) \\
&= \sum_{k=1}^{K} c_k.(\boldsymbol{x}_m^u - \mu_k)^T \Sigma_k (\boldsymbol{x}_m^u - \mu_k)
\end{aligned}
\tag{6}
$$

The Mahalanobis distance, obtained from one gaussian component, follows a β distribution (Ververidis & Kotropoulos, 2008), as shown in (7), where R is the random variable representing the Mahalanobis distance, M is the number of samples used for training, N is the dimensionality of the data, and $N < M$. It is important to note that, in this case as information is used from all clients, $M =$

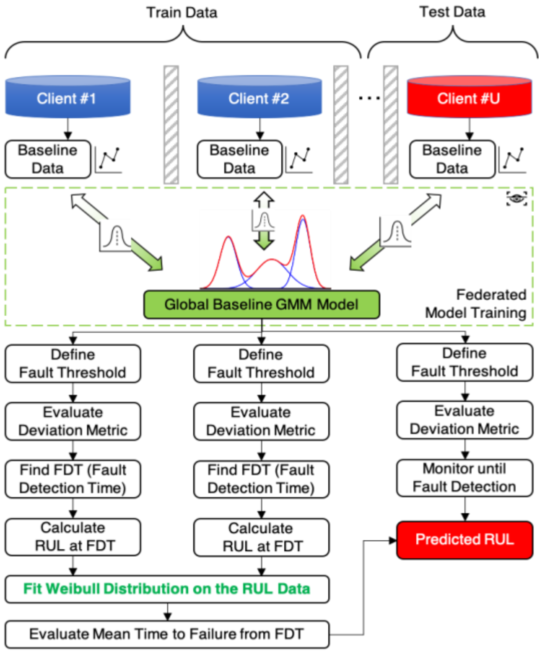
Train Data
Test Data
Client #1
Client #2
Client #U
Baseline Data
Baseline Data
Baseline Data
Global Baseline GMM Model
Federated Model Training
Define Fault Threshold
Define Fault Threshold
Define Fault Threshold
Evaluate Deviation Metric
Evaluate Deviation Metric
Evaluate Deviation Metric
Find FDT (Fault Detection Time)
Find FDT (Fault Detection Time)
Monitor until Fault Detection
Calculate RUL at FDT
Calculate RUL at FDT
Predicted RUL
Fit Weibull Distribution on the RUL Data
Evaluate Mean Time to Failure from FDT

Fig. 4 Proposed Methodology using Federated Global Baseline Modeling for Failure Prognosis

$\sum_{u=1}^{U} m_{baseline}^u$, where $m_{baseline}^u$ is the number of baseline samples in unit u.

$$\frac{M}{(M-1)^2}R \sim f_{Beta}\left(\frac{M}{(M-1)^2}r \,\middle|\, \frac{N}{2}, \frac{N-M-1}{2}\right) \qquad (7)$$

The cumulative distribution function is given by (8) where $I_x(a,b)$ is the incomplete beta function.

$$F_R(r) = I_{\frac{M}{(M-1)^2}r}\left(\frac{N}{2}, \frac{N-M-1}{2}\right) \qquad (8)$$

Thus, the threshold for fault detection is derived from the β Limit described by (9) at α = 0.99.

$$g'_u = \frac{(M-1)^2}{M}.F^{-1}\left(\alpha, \frac{N}{2}, \frac{N-M-1}{2}\right) + g_u^{drift} \qquad (9)$$

To adapt the threshold for a deviation score derived from multiple gaussian components, a scaling parameter g_u^{drift} is incorporated. This parameter is given by (10), where $\boldsymbol{\Theta}^u$ are the parameters of the local gaussian distribution of the unit u.

$$g_u^{drift} = \sum_{m=1}^{m_0^u} \frac{[d_u(\boldsymbol{x}_m^u \mid \boldsymbol{\Theta}) - d_u(\boldsymbol{x}_m^u \mid \boldsymbol{\Theta}^u)]}{m_{baseline}^u} \qquad (10)$$

The failure threshold established in (10) is used to detect incipient failures. Once the deviation metric in (6) exceeds the failure threshold, the incipient failure is detected, and the RUL is predicted. To simplify the discussion, the RUL is predicted as the mean time to failure (from incipient failure detection point to end of life) that is obtained from the training units. The training units are used to estimate the mean time to failure after an incipient failure is detected. For the training units, based on the fault detection time (FDT), the true remaining useful life (RUL) is calculated at FDT. This is the first instance at which RUL is predicted for a typical prognosis task. The RULs at FDT for each training unit are shared with the global server to fit into a Weibull distribution. The mean time to failure is then evaluated for predicting the time to failure at the first instance of detecting fault.

The baseline samples from the testing units also participate in the federated training of the global baseline model. The deviation metric as well as the fault detection threshold is obtained locally for each test unit. Every new incoming sample is monitored until the deviation metric crosses the fault detection threshold, at which point the remaining useful for the test unit is predicted.

4. CASE STUDIES

Two case studies are presented to showcase the performance of the proposed method. The first case study focuses on the performance of fitting a gaussian mixture model in a federated application compared to a traditional application on a simulated dataset. The second case study highlights the performance of the proposed method on remaining useful life prediction for Turbofan Engine Dataset.

4.1. Case Study 1: Simulative Study

Two simulated datasets are used from previous studies (Ververidis & Kotropoulos, 2008) to illustrate the clustering capability of the proposed federated expectation maximization algorithm. Dataset A is composed of eight partially overlapping distributions/clusters, with an equal number of i.i.d. samples from each distribution belonging to each client respectively. Dataset B is a set of three well-separated distributions/clusters, with samples from each distribution belonging to each of the three clients. Table 1 describes Dataset A and B, where μ_{true} and Σ_{true} represent the true mean and true covariances of the distribution from which the datasets are generated. m_0 represents the data samples for the respective cluster.

Two approaches are considered for parameter estimation of the simulated dataset – i) proposed federated expectation maximization, where no data is shared among the clients; and ii) traditional expectation maximization or centralized approach, where data from clients is aggregated together in a central server for parameter estimation.

Fig. 5 shows the performance of the proposed federated expectation maximization algorithm on Dataset A and B. The number of components is chosen based on the minimum AIC value achieved. Fig. 5(a) shows that the minimum AIC value is achieved for 8 components, which matches with the true number of clusters for Dataset A. Fig. 5(b) shows the probability density of the components over the dataset. The

Table 1 True Mean and Covariance of Distributions used to generate Datasets A and B

	Cluster #	$\boldsymbol{\mu_{true}}$	$\boldsymbol{\Sigma_{true}}$		**Cluster #**	$\boldsymbol{\mu_{true}}$	$\boldsymbol{\Sigma_{true}}$
Dataset A	1	[1.5, 0]	[0.01, 0; 0, 0.1]	**Dataset B**	1	[0, -2]	[2, 0; 0, 0.2]
	2	[1, 1]	[0.1, 0; 0, 0.1]		2	[0, 0]	[2, 0; 0, 0.2]
	3	[0, 1.5]	[0.1, 0; 0, 0.01]		3	[0, 2]	[2, 0; 0, 0.2]
	4	[-1, 1]	[0.1, 0; 0, 0.1]				
	5	[-1.5, 0]	[0.01, 0; 0, 0.1]				
	6	[-1, -1]	[0.1, 0; 0, 0.1]				
	7	[0, -1.5]	[0.1, 0; 0, 0.01]				
	8	[1, -1]	[0.1, 0; 0, 0.1]				

Table 2 Parameter Estimation for Simulated Data using Federated EM and Traditional EM for Dataset A

	Cluster #	**Samples distributed to Client #**	$\boldsymbol{m_0}$	$\boldsymbol{\mu_F}$	$\boldsymbol{\Sigma_F}$	$\boldsymbol{\mu_C}$	$\boldsymbol{\Sigma_C}$
Dataset A	1	1	200	[1.507, -0.01]	[0.013, 0.0; 0.0, 0.083]	[1.50, -0.008]	[0.013, 0.00; -0.001, 0.083]
	2	2	200	[0.978, 1.018]	[0.091, -0.005; -0.005, 0.088]	[0.978, 1.018]	[0.092, -0.01; -0.005, 0.088]
	3	3	200	[-0.05, 1.486]	[0.084, 0.003; 0.003, 0.01]	[-0.05, 1.486]	[0.084, 0.003; 0.003, 0.011]
	4	4	200	[-0.982, 1.01]	[0.11, 0.004; 0.004, 0.101]	[-0.98, 1.005]	[0.11, 0.003; 0.003, 0.101]
	5	5	200	[-1.47, -0.041]	[0.009, 0.003; 0.003, 0.967]	[-1.47, -0.041]	[0.01, 0.003; 0.03, 0.098]
	6	6	200	[-1.017, -1.0]	[0.082, 0.011; 0.011, 0.096]	[-1.016, -1.0]	[0.082, 0.011; 0.011, 0.1]
	7	7	200	[0.024, -1.48]	[0.964, 0.024; 0.024, 0.0111]	[0.023, -1.48]	[0.097, 0.002; 0.002, 0.011]
	8	8	200	[1.02, -0.97]	[0.096, -0.001; -0.001, 0.075]	[1.02, -0.978]	[0.097, -0.002; 0.0, 0.076]

Table 3 Parameter Estimation for Simulated Data using Federated EM and Traditional EM for Dataset B

	Cluster #	**Samples distributed to Client #**	$\boldsymbol{m_0}$	$\boldsymbol{\mu_F}$	$\boldsymbol{\Sigma_F}$	$\boldsymbol{\mu_C}$	$\boldsymbol{\Sigma_C}$
Dataset B	1	1, 2, 3	100	[0.062, -2.01]	[2.401, 0.078; 0.078, 0.213]	[0.061, -2.02]	[2.403, 0.077; 0.077, 0.21]
	2	1, 2, 3	100	[-0.08, -0.023]	[1.78, 0.037; 0.036, 0.29]	[-0.07, -0.025]	[1.77, 0.038; 0.038, 0.224]
	3	1, 2, 3	100	[0.014, 2.031]	[1.503, 0.061; 0.061, 0.1807	[0.0, 2.025]	[1.503, 0.062; 0.062, 0.179]

proposed federated approach can perform at par with the traditional approach as can be observed by comparing in Table 2. In Table 2, μ_F and Σ_F represent the estimated mean and covariance using federated expectation maximization, and μ_C and Σ_C represent the same parameters estimated from the centralized approach respectively for Dataset A.

Similarly, Fig. 5(c) shows the performance of the proposed method for Dataset B. In Dataset B, each client has samples from each of the three distributions. This can be compared to a case where each of the distribution can be considered as a particular asset operating under different conditions with multiple clients. The data from each cluster is randomly distributed across every client. This represents the case of non-uniform availability of data across clients. Fig. 5(c) and Fig. 5(d) show that the optimal number of components or clusters for dataset B is correctly estimated as three with the presented probability density spread. Table 3 shows the estimated parameters of the clusters for Dataset B using both approaches and illustrates that the estimation by the federated EM approach matches with the estimation by the traditional EM approach. With this global model trained in a federated way, every client organization has access to more knowledge of the dataset distribution irrespective of the local availability of the data, preserving data privacy. For data-poor clients, this can bring benefit through learned models without having to wait for enough data to be collected.

4.2. Case Study 2: Aero-engine prognosis

4.2.1. Data and Problem Description

An implementation of the proposed algorithm is shown for collaborative fault prognosis using the Turbofan Engine Degradation Dataset (Saxena & Goebel, 2008). The data is generated using the MATLAB-based software called C-MAPPS, which is designed to simulate the behavior of commercial turbofan engines and is widely used in the literature. The turbofan engines simulated by C-MAPPS are formed by several inter-dependent sub-systems, resembling the mechanisms typically present in industrial machinery. One of the sub-systems included is a limiter, which prevent machines from exceeding pre-set tolerances. Simulation parameters include environmental, control, and failure parameters, including a set of health-parameter inputs that are designed to simulate deterioration and fault. Time-series variables that represent parameters such as fan speed, temperatures at various locations in the system, engine pressure ratio, etc. are recorded over the life of the engine. A comprehensive diagram and description of the turbine simulated by C-MAPPS along with the list of variables

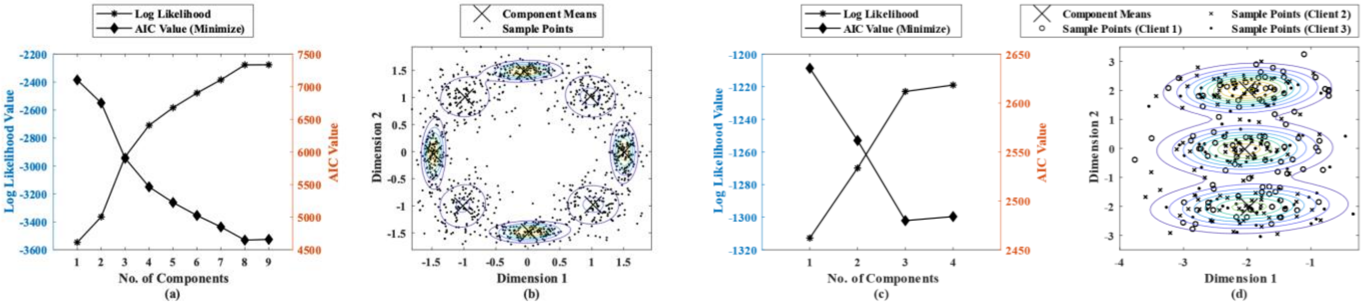

Fig. 5 Performance of proposed federated EM algorithm on simulated Dataset A (a) and (b) and Dataset B (c) and (d).

recorded can be found in (Saxena & Goebel, 2008). The dataset used in this work considers operation at sea level and degradation of High-Pressure Compressor (HPC) and consists of 100 multi-variate run-to-failure time series. Each time series represents a different engine unit. Each engine starts with different degrees of initial wear and manufacturing variation which is unknown to the user. This wear and variation are considered normal, i.e., it is not considered as a fault condition. The recorded data is contaminated with sensor noise. Measurements from seven sensors – 2, 3, 4, 7, 11, 12, and 15 – are used in the analysis based on the past literature (Wang, Yu, Siegel & Lee, 2008). The engine is operating normally at the start of each time series and develops a fault at some unknown point during the operation. The fault grows in magnitude until system failure, and the number of operational cycles until failure is recorded.

The problem addressed in this case study is to predict the remaining useful life of the engine units at the point of fault detection. The methodology proposed in Figure 4 is followed for this case study. Two experimental settings are used for evaluation. In Experiment Setting I the entire dataset is divided into training and testing units, wherein the failure time of the training units is known, and the failure time of the testing units is unknown. The training units are used to estimate the mean time to failure from the point of first fault detection, and the accuracy of the prediction is evaluated on the testing units using root mean square error (RMSE). An iteration consists of using 75% of the total units randomly selected for training, and the remaining 25% used for testing, with a four-fold cross validation. Each iteration is repeated ten times. In Experiment Setting II all the 100 units are used as training units to estimate the bounds of the mean time to failure. The difference between the upper and lower bound of the estimated mean time to failure is used as the metric of performance.

4.2.2. Performance Benchmarking

The performance of the proposed federated learning approach is compared with the following traditional approaches. The fault prognosis architecture remains the same for each of the following benchmarking methods except for changes in the data sharing strategy, baseline modeling techniques, and fault detection metrics. All methods labeled Stand-Alone (SA) make use of only the local data to train a local baseline model representing the scenario where no data sharing is allowed. In these methods, each unit has their separate local baseline models instead of a global baseline model.

1. *No Baseline*: A Weibull distribution is directly fitted to the training dataset for predicting time to failure without using a baseline model to detect the point of fault occurrence.
2. *GMM-SA*: This method is a stand-alone (SA) implementation of training the local gaussian mixture model for each unit. It does not collaborate with other units either through data or model for training the baseline model.
3. *SOM-SA*: This method uses only the local data to train a self-organizing map for fault detection. The baseline Minimum quantization error is used as a deviation metric, for which a 3-sigma limit is used as a threshold.
4. *SVM-SA*: A one-class support vector machine is used to train the local baseline model. A radial basis function kernel is used with a lower bound of the fraction of support vectors relative to the total number of training examples as 0.05. A fault is detected when consecutive samples equal to the length on the moving window are labeled as outliers.

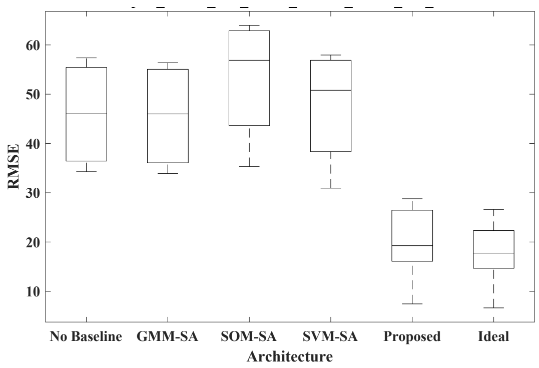

Fig. 6 Performance of various methods using RMSE on RUL Prediction using Experiment Settings I

5. *Proposed*: This is the proposed method that uses federated training of a global gaussian mixture model for fault detection as described in Fig. 4.
6. *Ideal*: This method trains a global gaussian mixture model by allowing sharing of data. It represents the traditional method for model training and signifies the best possible scenario for fault detection using gaussian mixture model.

4.2.3. Results and Discussion

The root mean square error of all six methods for 100 units using Experiment Settings I is shown in Fig. 6 as a box plot. $m_{baseline}^{u}$ is considered as 10 samples. The proposed and the ideal methods perform significantly better than any of the stand-alone approaches or when no baseline model is considered. This highlights the huge benefit of performing collaborative prognosis. These results represent a scenario where each individual unit may belong with a different client organization or user and data sharing across organizations may not be allowed. Using the proposed federated method for collaborative fault prognosis allows achieving performance as good as performing fault prognosis traditionally using data sharing. Using the proposed method, both the mean and the spread of the RMSE is better than SA approaches and at par with the ideal scenario.

Since the performance of stand-alone methods can be affected by the number of baseline samples, $m_{baseline}^{u}$, available for model training, Fig. 7 shows the average RMSE of each method by varying the number of baseline samples per unit. The model with no baseline is unaffected as it does not use a baseline model for fault detection. The performance of the stand-alone method improves as the number of baseline samples per unit increases, but only until a certain level. After about the value of $m_{baseline}^{u}$ = 70, the performance either worsens or remains unaffected for every stand-alone method. However, the proposed approach remains consistently better and unaffected by the local change in $m_{baseline}^{u}$. This again highlights the benefit of collaborative learning, where the shortcomings of one organization can be overcome by the strength of other organizations. This advantage is most significant for units with less amount of baseline data. Even for units with a high amount of baseline data, the proposed approach remains better than stand-alone approaches and at par with the ideal scenario.

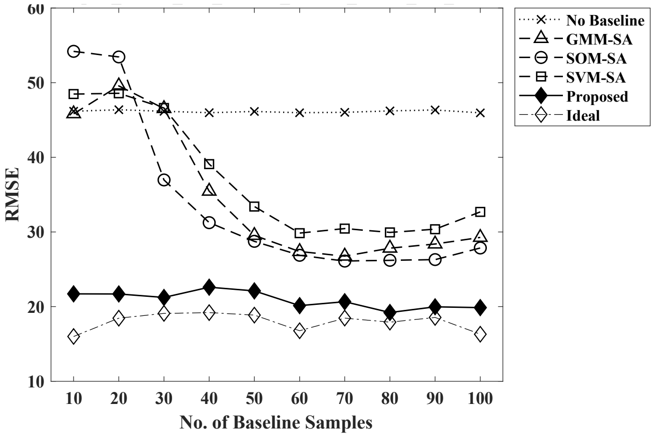

Fig. 7 Performance of various methods using Average RMSE on RUL Prediction by varying $m_{baseline}^{u}$ using Experiment Settings I.

For effective collaboration for fault prognosis, the number of collaborating organizations or collaborating users plays an important role. Moreover, there can scenarios where data sharing may be possible among units that belong to one organization. For organizations with a huge fleet and the possibility of data sharing, using federated learning may not always be the solution for improved performance. But for an organization with one or few units, as is the case in Fig. 6 and Fig. 7, it can be very beneficial to collaborate with a large number of organizations. Thus, it becomes important to make recommendations on where federated learning can be

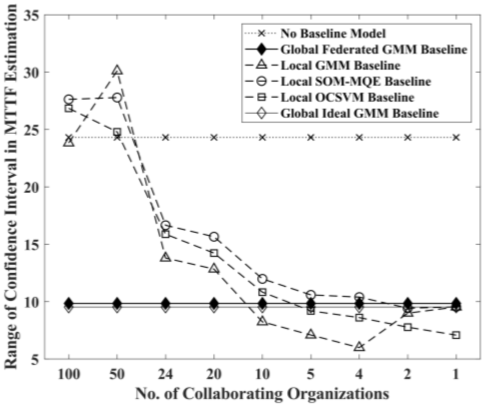

Fig. 8 Effect of Collaborating Organizations on Robustness of Failure Prediction for $m_{baseline}^{u}$ = 10 using Experiment Settings II

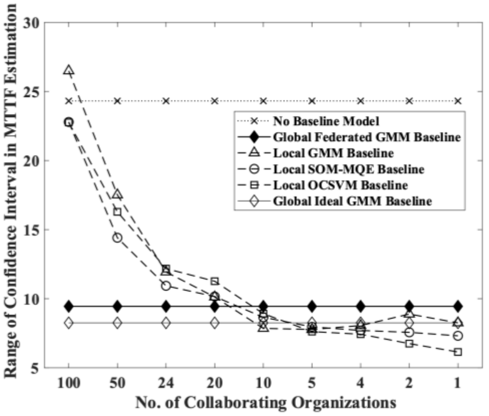

Fig. 9 Effect of Collaborating Organizations on Robustness of Failure Prediction for $m_{baseline}^{u}$ = 20 using Experiment Settings II

beneficial. Fig. 8 and Fig. 9 present the performance of proposed methods for decreasing the number of collaborating organizations and use Experiment Settings II. 100 units are distributed equally over the number of organizations. For example, for 100 collaborating organizations, each organization has 1 unit, whereas, for 2 collaborating organizations, each organization has 50 units. The case of only 1 organization represents the ideal scenario where data sharing across all units is allowed. For this analysis, the standalone notation is updated to local notation, as data sharing within the organization is allowed.

The value of $m^{u}_{baseline}$ for the results reported in Fig. 8 is 10 and in Fig. 9 is 20. From both the figures four major inferences can be made. First, the performance of the proposed method is consistent and almost at par with ideal method irrespective of the number of collaborating organizations. This further validates the competency of the proposed method allowing minimum deviation from ideal scenario even with having data barriers for privacy protection in place across units. Secondly, the benefit of federated collaborative learning is dependent on the number of collaborating organizations. As the number of collaborating organizations decrease and the number of units within an organization increase, it becomes less favorable to collaborate. Thirdly, by comparing Fig. 8 and Fig. 9, for collaborating organizations greater than 10, the RMSE value decreases significantly with the increase in the value of $m^{u}_{baseline}$. However, the performance of local models becomes better than the collaborative model only for more than 10 collaborating organizations. Thus, collaboration is most beneficial for units with a lesser number of baseline samples. Finally, the performance of the gaussian mixture model can be affected by the increase in the size of data. On the other hand, the performance of SOM and OCSVM follow a consistent trend of reduction in RMSE as the number of collaborating organizations decreases. This happens as the tendency of the expectation maximization algorithm to converge at a local-optimum increases with an increase in data size. Advanced versions of gaussian mixture model training such as (Balakrishnan, Wainwright & Yu, 2017; Zhao, Li & Sun, 2020) could be further developed to be implemented in federated settings to address this issue and forms the future scope of this work.

5. CONCLUSION

A systematic analytical methodology to build collaborative prognostic models from decentralized databases addressing limitations of data-privacy and local label scarcity is proposed. The methodology facilitates effective learning of asset health conditions for data-scarce organizations by collaborating with other organizations preserving data privacy. Collaboration is more beneficial when data for common assets is sparsely distributed across organizations. This is most suitable for a servitization model for Overall Equipment Manufacturers who sell to multiple organizations. While the presented work assumes same type of fault (labels) in the distributed datasets for prognosis, future work can consider development of methods addressing variations in fault types, machine operation conditions, etc. This is more likely to occur in real life scenarios as model aggregation happens across different organizations. Such variations make structured datasets with same labels even more scarce and distributed leading to the overall data distribution becoming non i.i.d.. For such situations, federated learning can prove even more valuable as advanced algorithms are developed to train models from non i.i.d. data distributions under this framework.

REFERENCES

Balakrishnan, S., Wainwright, M. J., & Yu, B. (2017). Statistical guarantees for the EM algorithm: From population to sasmple-based analysis. *Annals of Statistics*, vol. 45, no. 1, pp. 77-120.

Baines, T. S., Lightfoot, H. W., Evans, S., Neely, A., Greenough, R., Peppard, J., Roy, R., Shehab, E., Braganza, A., Tiwari, A., & Alcock, J. R. (2007). State-of-the-art in product-service systems, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 221, no.10, pp. 1543-1552.

Bisong E. (2019) Batch vs. Online Learning. In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4470-8_15

Dhada, M., Parlikad, A., & Palau, A.S. (2020). Federated learning for collaborative prognosis. In *International Conference on Precision, Meso, Micro, and Nano Engineering* doi: 10.17863/CAM.50577

Diaz-Rozo, J., Bielza, C., & Larrañaga, P. (2020). Machine-tool condition monitoring with Gaussian mixture models-based dynamic probabilistic clustering. *Engineering Applications of Artificial Intelligence*, vol. *89*, pp. 103434.

Jin, C., Zhang, Y., Balakrishnan, S., Wainwright, M. J., & Jordan, M. (2016). Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences. *arXiv preprint arXiv:1609.00978*.

Jin, R., Huang, Y., He, X., Wu, T., & Dai, H. (2020). Stochastic-sign SGD for federated learning with theoretical guarantees. *arXiv preprint arXiv:2002.10940*.

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

Kurban, H., Jenne, M., & Dalkilic, M. M. (2017). Using data to build a better EM: EM* for big data. *International Journal of Data Science and Analytics*, vol. 4, no. 2, pp. 83-97.

Lapira, E. R. (2012). Fault detection in a network of similar machines using clustering approach (Doctoral dissertation, University of Cincinnati).

Lin, Y., Liu, K., Byon, E., Qian, X., Liu, S., & Huang, S. (2017). A collaborative learning framework for estimating many individualized regression models in a heterogeneous population. *IEEE Transactions on Reliability*, vol. 67, no. 1, pp. 328-341.

Lee, J., Davari, H., Singh, J., & Pandhare, V. (2018). Industrial Artificial Intelligence for industry 4.0-based manufacturing systems, *Manufacturing letters*, vol. 18, pp. 20-23.

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical systems and signal processing*, vol. 104, pp. 799-834.

Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, vol. *37*, no. 3), 50-60.

Maroli, J. M., Özgüner, Ü., & Redmill, K. (2019, June). A Framework for Automated Collaborative Fault Detection in Large-Scale Vehicle Networks. *2019 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1923-1927). IEEE.

McKinsey D. (2016). Industry 4.0 after the initial hype. Where manufacturers are finding value and how they can best capture it.

Ng, Y. S., & Srinivasan, R. (2010). Multi-agent based collaborative fault detection and identification in chemical processes. *Engineering Applications of Artificial Intelligence*, vol. 23, no. 6, pp. 934-949.

Palau, A. S. Dhada, M. H., Bakliwal, K., & Parlikad, A. K. (2019). An Industrial Multi Agent System for real-time distributed collaborative prognostics, *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 590-606.

Pan, S.J. & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), pp.1345-1359.

Pimentel, M. A., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, vol. 99, pp. 215-249.

Regulation (2016) (EU) 2016/679 of the European Parliament and of the Council. *REGULATION (EU)*, vol. *679*.

Rosero, R. H. L., Silva, C., & Ribeiro, B. (2020, July). Remaining useful life estimation in aircraft components with federated learning. *PHM Society European Conference*, Vol. 5, No. 1, pp. 9-9.

Saxena, A., & Goebel, K. (2008). Turbofan Engine Degradation Simulation Data Set, *NASA Ames Prognostics Data Repository*, NASA Ames Research Center, Moffett Field, CA (http://ti.arc.nasa.gov/ project/prognostic-data-repository)

Ververidis, D., & Kotropoulos, C. (2008). Gaussian mixture modeling by exploiting the Mahalanobis distance. *IEEE transactions on signal processing*, vol. *56, no.* 7, pp. 2797-2811.

Wang, T., Yu, J., Siegel, D., & Lee, J. (2008, October). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *2008 international conference on prognostics and health management* (pp. 1-6). IEEE.

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications, *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1-19.

Yi, S., Li, C., & Li, Q. (2015). A survey of fog computing: concepts, applications and issues, *In Proceedings of the 2015 workshop on mobile big data*, (37-42).

Zhao, R., Li, Y., & Sun, Y. (2020). Statistical convergence of the EM algorithm on Gaussian mixture models. *Electronic Journal of Statistics*, vol. *14, no.* 1, pp. 632-660.

Zhao, Y., Li, D., Lu, T., Lv, Q., Gu, N., & Shang, L. (2020). Collaborative fault detection for large-scale photovoltaic systems. *IEEE Transactions on Sustainable Energy*, vol. *11, no.* 4, pp. 2745-2754.