Technometrics

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/utch20

Federated Multi-Output Gaussian Processes

Seokhyun Chung & Raed Al Kontar

To cite this article: Seokhyun Chung & Raed Al Kontar (2024) Federated Multi-Output Gaussian Processes, Technometrics, 66:1, 90-103, DOI: 10.1080/00401706.2023.2238834

To link to this article: https://doi.org/10.1080/00401706.2023.2238834

TECHNOMETRICS
2024, VOL. 66, NO. 1, 90–103
https://doi.org/10.1080/00401706.2023.2238834

Taylor & Francis
Taylor & Francis Group

Federated Multi-Output Gaussian Processes

Seokhyun Chung[a] and Raed Al Kontar[b]

[a]Systems & Information Engineering, University of Virginia, Charlottesville, VA; [b]Industrial & Operations Engineering, University of Michigan, Ann Arbor, MI

ABSTRACT

Multi-output Gaussian process (MGP) regression plays an important role in the integrative analysis of different but interrelated systems/units. Existing MGP approaches assume that data from all units is collected and stored at a central location. This requires massive computing and storage power at the central location, induces significant communication traffic due to raw data exchange, and comprises privacy of units. However, recent advances in Internet of Things technologies, which have tremendously increased edge computing power, pose a significant opportunity to address such challenges. In this article, we propose `FedMGP`, a general federated analytics (FA) framework to learn an MGP in a decentralized manner that uses edge computing power to distribute model learning efforts. Specifically, we propose a hierarchical modeling approach where an MGP is built upon shared global latent functions. We then develop a variational inference FA algorithm that overcomes the need to share raw data. Instead, collaborative learning is achieved by only sharing global latent function statistics. Comprehensive simulation studies and a case study on battery degradation data highlight the superior predictive performance and versatility of `FedMGP`, achieved while distributing computing and storage demands, reducing communication burden, fostering privacy, and personalizing analysis.

ARTICLE HISTORY
Received May 2022
Accepted June 2023

KEYWORDS
Federated analytics; Internet of federated things; Multi-output Gaussian processes; Variational inference

1. Introduction

The multi-output Gaussian process (MGP), also known as *co-kriging* (Ver Hoef and Barry 1998), is a popular tool for integrative analysis of output from different but interrelated systems/units. It is the extension of Gaussian processes (GP) to vector-valued outputs. The key idea is to construct a large covariance matrix that defines covariances both within and across different outputs. As a result, an MGP can leverage commonalities across related units, often resulting in enhanced learning and improved predictive performance over independent model learning within each unit. MGPs naturally inherit the advantageous properties of GPs such as nonparametricity and uncertainty quantification capability (Handcock and Stein 1993). With such promising merits and recent advances in GP that allow for scaling to large-size data (e.g., Guhaniyogi and Banerjee 2018; Chen et al. 2020), MGP has seen great success in various domains. A range of examples can be found in geostatistics (e.g., Gotway and Young 2002; Li and Zimmerman 2015), reliability engineering (e.g., Kontar et al. 2018), urban planning (e.g., Bae et al. 2018), computer simulation (e.g., Mak et al. 2018; Huang and Gramacy 2021), additive manufacturing (e.g., Chen et al. 2021), healthcare (e.g., Cheng et al. 2020; Chung, Al Kontar, and Wu 2022), among many others.

Despite many success stories, existing studies on MGP assume that data from all units is processed centrally; that is, data across all units is stored and processed at the same central location. For instance, consider a telematics system that performs fleet management based on battery degradation data collected from multiple vehicles. In a traditional centralized system, every vehicle should send its degradation signals to the central server, then the central server estimates the MGP using collected signals from the vehicles, and finally, the server distributes results to the vehicle users (e.g., the predicted residual battery life). This is illustrated in Figure 1(a). This centralized learning where all data from all units is located in one place has long been the underlying assumption when modeling and inferring an MGP.

However, recent advances in semiconductor technologies have facilitated the deployment of edge units equipped with highly compact AI chips with remarkable computing capabilities. For example, in telematics systems nowadays, furnishing vehicles with computing resources has become increasingly straightforward. This evolution in computing resources at the edge paves the way for a new analytics paradigm within the Internet of Things (IoT) built upon local computations and the decentralization of data analysis. In this paradigm, units exploit their computing capabilities to transfer part of the model learning to the edge; where the data is actually created. This new IoT system characterized by edge computing power, has been coined the Internet of Federated Things (IoFT) (Kontar et al. 2021) and is illustrated in Figure 1(b). Within IoFT, Federated Analytics (FA) defines the decentralized data analytics approach in IoFT that enables collaborative

CONTACT Raed Al Kontar *alkontar@umich.edu* Industrial & Operations Engineering, University of Michigan, Ann Arbor, MI.
Supplementary materials for this article are available online. Please go to *www.tandfonline.com/r/TECH*.

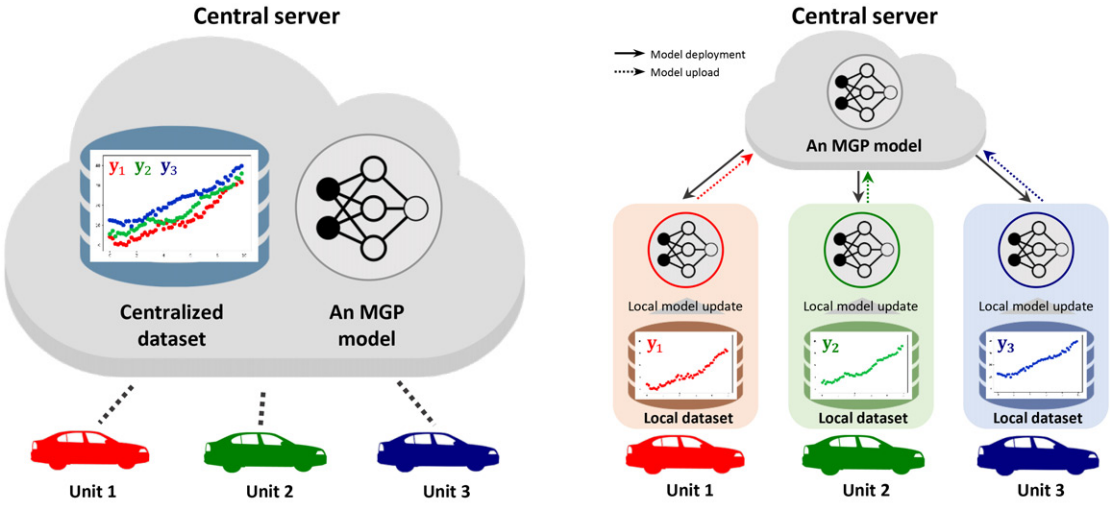

Figure 1. Illustrative comparison of a centralized system and a federated system.

model learning. The word "federated" here refers to some extent of localization and autonomy conferred on units at the edge, resulting from the decentralization of data analytics efforts.

Needless to say, FA resolves fundamental issues arising in traditional centralized IoT. To name a few, the ability to perform local computations allows (i) improved privacy, as only focused updates from local computations needed for collaborative model learning need to be shared, in contrast to sharing raw data in centralized systems (ii) reduced computation and storage needs imposed on the central server, as the server now mainly serves an integration point of shared knowledge (iii) reduced latency and communication traffic, as there is no massive raw data exchange between the server and units (iv) faster alerts and decision, as a local model now exists for immediate action. The MGP, as an intrinsic approach for collaborative model learning across different units, can reap immense benefits from FA. However, the current literature on MGP is still limited to centralized analytics. This study aims to fill the gap existing between FA and MGP.

Specifically, we propose `FedMGP`, a general framework for MGP-based federated integrative analysis where correlated units collaboratively learn an MGP model without any raw data exchange. The fundamental intuition is to assume that outputs of different units are generated from a set of common global latent functions. By collaboratively learning those latent functions, units can indirectly borrow strength from each other while circumventing the need to share raw data. Instead, only knowledge on the global functions needs to be shared. Building on this intuition, we construct a hierarchical probabilistic model where outputs are assumed to follow independent GPs conditional on shared latent processes. This hierarchical approach allows for characterizing both dependencies across units and their unique features, but at the same time, poses a significant challenge that all data needs to be available centrally during parameter estimation and posterior inference. To address this issue, we introduce a variational inference (VI) approach which approximates the posterior distribution over the latent processes. We show that our VI-based approach, in turn, provides a variational lower bound suitable for FA. This mitigates enormous computational demands entailed by centralized MGPs built on extremely large datasets; a major inhibitor in the broad use of existing MGPs. It also allows both model learning and prediction without raw data sharing at any stage.

In addition to the contributions above our study brings contributions to FA. We develop a new methodology of FA that provides a natural Bayesian interpretation and estimates a predictive posterior distribution, allowing us to obtain quantified predictive uncertainty. This is particularly important for subsequent decision-making and prescriptive analytics. Another interesting contribution is that our federated variational approach results in a personalized prediction for each unit. Personalization is essential to capture unique behaviors of units, often found in practice where units are operated in different environments.

We assess our framework using simulated and real-world data. We evaluate prediction accuracy for various scenarios of IoFT such as nonparametric signals, unstable communication, participation of many units, and the presence of anomalous units. In the case study, we consider an application in reliability engineering where our model is applied to estimating the degradation curves of lithium-ion batteries. Our model is compared with a centralized MGP model, a two-step train-then-personalize approach, a module-based approach, and separate models independently inferred by each unit. Our results highlight the effectiveness of the proposed approach and its ability to transfer knowledge across units, achieved in a way that distributes computing and storage demands, reduces communication burden, enhances privacy, and personalizes analysis.

The remainder of the article is organized as follows. Section 2 provides an overview of FA and critical challenges in its direct application to estimating an MGP. Section 3 introduces our proposed approach. Section 4 reviews relevant literature that aims to distribute computing efforts in GPs, categorized as module-based and FA-based approaches. Section 5 examines the proposed approach using simulation data. Section 6 applies our method to a real-world application in reliability engineering.

Section 7 concludes the article with a discussion on possible future directions.

2. Preliminaries: Centralized MGP, FA, and Challenges

We start by building notation. Consider M units indexed by $\mathcal{M} = \{1, \ldots, M\}$. A unit $m \in \mathcal{M}$ collects N_m observations, denoted by $\mathcal{D}_m \equiv (\mathbf{X}_m, \mathbf{y}_m)$, with input $\mathbf{X}_m = [\mathbf{x}_{m,n}]_{n=1,\ldots,N_m}^\top \in \mathbb{R}^{N_m \times d}$ and output $\mathbf{y}_m = [y_{m,n}]_{n=1,\ldots,N_m}^\top \in \mathbb{R}^{N_m \times 1}$. We also denote $\mathbf{X} = [\mathbf{X}_m^\top]_{m=1,\ldots,M}^\top \in \mathbb{R}^{N \times d}$ and $\mathbf{y} = [\mathbf{y}_m^\top]_{m=1,\ldots,M}^\top \in \mathbb{R}^{N \times 1}$ as the concatenated inputs and outputs from all units where $N = \sum_{m \in \mathcal{M}} N_m$.

In Section 2.1, we review an MGP in a centralized data environment. In Section 2.2, we introduce a general framework for FA and its extension to personalized FA. In Section 2.3, we pose critical challenges arising when employing an MGP under federated settings.

2.1. Centralized MGP

Suppose that the output of unit $m \in \mathcal{M}$ is expressed as

$$y_{m,n} = f_m(\mathbf{x}_{m,n}) + \epsilon_m \quad \text{for} \quad n \in \{1, \ldots, N_m\},$$

where $f_m(\cdot)$ denotes the true output function and ϵ_m denotes additive noise for unit m. An MGP is built upon defining a shared covariance between observations across all outputs. To do so, define $\mathbf{f} = [\mathbf{f}_m^\top]_{m=1,\ldots,M}^\top$ with $\mathbf{f}_m = [f_{m,n}]_{n=1,\ldots,N_m}^\top = [f_m(\mathbf{x}_{m,n})]_{n=1,\ldots,N_m}^\top$ to collectively denote underlying true values of the output at the observed input. Also, let $\mathbf{C}_{\mathbf{f}_m,\mathbf{f}_{m'}} \in \mathbb{R}^{N_m \times N_{m'}}$ denote a (cross-) covariance matrix calculated from a covariance function $c_{f_m,f_{m'}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_{f_m,f_{m'}}) = \text{cov}(f_m(\mathbf{x}), f_{m'}(\mathbf{x}'))$ parameterized by $\boldsymbol{\theta}_{f_m,f_{m'}}$. An MGP is then defined over $\mathbf{f}$ as

$$\mathbf{f} = \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_M \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \mathbf{C}_{\mathbf{f}_1,\mathbf{f}_1} & \cdots & \mathbf{C}_{\mathbf{f}_1,\mathbf{f}_M} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{\mathbf{f}_M,\mathbf{f}_1} & \cdots & \mathbf{C}_{\mathbf{f}_M,\mathbf{f}_M} \end{pmatrix}\right) := \mathcal{N}(\mathbf{0}, \mathbf{C}). \tag{1}$$

Assuming iid Gaussian noises $\epsilon_m \sim \mathcal{N}(0, \sigma_m^2)$ for $m \in \mathcal{M}$, the observational model is given as

$$p(\mathbf{y}|\mathbf{X}; \boldsymbol{\theta}_{\text{full}}, \boldsymbol{\sigma}) = \int p(\mathbf{y}|\mathbf{f}; \boldsymbol{\sigma}) p(\mathbf{f}|\mathbf{X}; \boldsymbol{\theta}_{\text{full}}) d\mathbf{f} = \psi(\mathbf{y}; \mathbf{0}, \mathbf{C} + \boldsymbol{\Sigma}) \tag{2}$$

where $\psi(\cdot; \boldsymbol{\mu}, \boldsymbol{\Xi})$ is a multivariate Gaussian density with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Xi}$. Also, $\boldsymbol{\theta}_{\text{full}} = \{\boldsymbol{\theta}_{f_m,f_{m'}}\}_{m,m'=1}^{M,M}$, $\boldsymbol{\sigma} = \{\sigma_m\}_{m=1}^M$ and $\boldsymbol{\Sigma} = \text{bdiag}(\sigma_m^2 \mathbf{I}_{N_m})_{m=1}^M$ where $\mathbf{I}_a$ denotes an $a \times a$ identity matrix and $\text{bdiag}(\mathbf{A}_i)_{i=1}^I$ is a block diagonal matrix with $\mathbf{A}_1, \ldots, \mathbf{A}_I$ on the diagonal.

Model estimation in an MGP has a rich history. Most often it is done by maximizing the marginal log-likelihood (2) in terms of $\{\boldsymbol{\theta}_{\text{full}}, \boldsymbol{\sigma}\}$, that is, $\max_{\boldsymbol{\theta}_{\text{full}},\boldsymbol{\sigma}} \log \psi(\mathbf{y}; \mathbf{0}, \mathbf{C} + \boldsymbol{\Sigma})$. Many iterative optimization algorithms such as gradient-based methods can be used. Whichever algorithm is chosen, evaluating either (2) or its gradient is usually required at each iteration. This is readily done in a centralized regime as $c_{f_m,f_{m'}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_{f_m,f_{m'}})$ can be directly evaluated when all datasets $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$ are centrally available.

Upon parameter estimation and given a new input observation $\mathbf{x}_m^*$ at unit m, the predictive distribution of the corresponding output y_m^* is derived as

$$\begin{aligned} p(y_m^*|\mathbf{x}_m^*, \mathbf{X}, \mathbf{y}; \boldsymbol{\theta}_{\text{full}}, \boldsymbol{\sigma}) &= \int p(y_m^*|\mathbf{f}, \mathbf{x}_m^*, \mathbf{X}; \boldsymbol{\theta}_{\text{full}}, \boldsymbol{\sigma}) p(\mathbf{f}|\mathbf{X}, \mathbf{y}; \boldsymbol{\theta}_{\text{full}}) d\mathbf{f} \\ &= \psi(y_m^*; \mathbf{c}^{*\top}(\mathbf{C} + \boldsymbol{\Sigma})^{-1}\mathbf{y}, \text{cov}(\mathbf{x}_m^*, \mathbf{x}_m^*) \\ &\quad - \mathbf{c}^{*\top}(\mathbf{C} + \boldsymbol{\Sigma})^{-1}\mathbf{c}^* + \sigma_m^2) \end{aligned} \tag{3}$$

where $\mathbf{c}^* \in \mathbb{R}^{N \times 1}$ is a vector composed of covariances between $\mathbf{X}$ and $\mathbf{x}_m^*$ and $\text{cov}(\mathbf{x}_m^*, \mathbf{x}_m^*) = c_{f_m,f_m}(\mathbf{x}_m^*, \mathbf{x}_m^*; \boldsymbol{\theta}_{f_m,f_m})$.

2.2. Federated Analytics (FA)

Most FA approaches in the literature focus on deep neural networks (DNN) and thereby are built upon an empirical risk minimization (ERM) framework (Vapnik 1991). Therefore, we start our overview of FA by explaining ERM over multiple units. Unlike a centralized regime in the previous section, we consider a decentralized regime that keeps data $\mathcal{D}_m$ stored locally at unit m. Hereon, we will use the terms client and unit interchangeably. We will also start with a generic framework for FA and in the following sections highlight how our MGP treatment can be synergized with such a framework.

Let $f(\cdot; \boldsymbol{\theta})$ denote a global model to be learned parameterized by $\boldsymbol{\theta}$. In ERM, we aim to find $\boldsymbol{\theta}$ that minimizes the global empirical risk of the model f across all data $\mathcal{D} = \{\mathcal{D}_m\}_{m \in \mathcal{M}}$. This is expressed as

$$\min_{\boldsymbol{\theta}} \mathcal{R}(\boldsymbol{\theta}; \mathcal{D}) = \min_{\boldsymbol{\theta}} \sum_{m \in \mathcal{M}} p_m \mathcal{R}_m(\boldsymbol{\theta}; \mathcal{D}_m), \tag{4}$$

where $\mathcal{R}_m(\boldsymbol{\theta}; \mathcal{D}_m)$ is the local empirical risk for unit $m \in \mathcal{M}$ and p_m is a weight often equally distributed over units ($\frac{1}{M}$) or proportional to the number of data points ($\frac{N_m}{N}$). Under some loss function, ℓ, the local empirical risk of the unit m is

$$\mathcal{R}_m(\boldsymbol{\theta}; \mathcal{D}_m) = \frac{1}{N_m} \sum_{n=1}^{N_m} \ell(f(\mathbf{x}_{m,n}; \boldsymbol{\theta}), y_{m,n}).$$

In a centralized database where all data $\mathcal{D}$ is accessible, we can directly optimize (4). On the other hand, in a federated setting, each client m can only evaluate their own risk function, and the central server does not have the data from the clients. To address this, FA takes a naturally distributed route illustrated in Algorithm 1 to learn $\boldsymbol{\theta}$. The detailed procedure is rather intuitive. The central server broadcasts a global model to each unit. Each unit m executes `client_update` to update the deployed model using its local data $\mathcal{D}_m$ and finds a locally updated model $\boldsymbol{\theta}_m^{\text{local}}$. The unit then sends back $\Delta_m^{\text{local}} = \boldsymbol{\theta} - \boldsymbol{\theta}_m^{\text{local}}$ called *local pseudo-gradient*. This procedure is referred to as a local update, where local computing power is exploited. Then the central server runs `cloud_update` to integrate all *global pseudo-gradient*. This procedure is iterated until a termination condition is satisfied. Here an iteration of FA is referred to as a communication round.

FA was recently brought to the vanguard of data analytics after the release of the seminal work by McMahan et al.

Algorithm 1 A generic framework for FA

Input: $\boldsymbol{\theta}$ (the initial global model); E (local steps); η (the global step size)

while not a terminating condition is satisfied **do**
  The central server broadcasts $\boldsymbol{\theta}$ to all clients
  for each client $m \in \mathcal{M}$ **do**
    Updates the deployed model:
$$\boldsymbol{\theta}_m^{\text{local}} \leftarrow \texttt{client\_update}(\boldsymbol{\theta}; \mathcal{D}_m, E)$$
    Calculates a local pseudo-gradient: $\Delta_m^{\text{local}} \leftarrow \boldsymbol{\theta} - \boldsymbol{\theta}_m^{\text{local}}$
    Uploads Δ_m^{local} to the central server
  end for
  The central server calculates the global pseudo-gradient:
$$\Delta \leftarrow \texttt{cloud\_update}(\{\Delta_m^{\text{local}}\}_{m\in\mathcal{M}})$$
  Update the global model: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\Delta$
end while

(2017), a research team at Google. In the paper, they proposed `FedAvg`, a communication-efficient approach to train a DNN under a decentralized data environment. `FedAvg` can be viewed as a special case of FA, where `client_update` is done through E iterates of stochastic gradient descent (SGD) and `cloud_update` averages local pseudo-gradients by $\Delta = \sum_{m\in\mathcal{M}} p_m \Delta_m^{\text{local}}$ and updates the global model with $\eta = 1$. Since first proposed in 2017, `FedAvg` has become the standard algorithm for FA thanks to its simplicity and robust performance. Many other methodologies extend `FedAvg` to deal with challenges in federated settings. For instance, approaches to deal with data heterogeneity across units (e.g., Zhao et al. 2018; Sattler et al. 2019; Zhu et al. 2021), to ensure fair predictive performance (e.g., Li et al. 2020b; Yue, Nouiehed, and Al Kontar 2022), to mitigate stragglers with limited bandwidth or computing capability (e.g., Li et al. 2020a; Park et al. 2021), and to defend against potential backdoor attacks (e.g., Sun et al. 2019; Bagdasaryan et al. 2020; Xie et al. 2021) were proposed. We refer the reader to an in-depth review of FA by Kontar et al. (2021).

2.2.1. Personalized FA

A commonality of the FA approaches above is that one global model is learned to fit all clients. In contrast, one may also create tailor-made models that account for client heterogeneity while still leveraging common knowledge across clients. This is exactly what MGP does as predictions in an MGP are output-specific. Along this line, learning a personalized model has been actively investigated in FA. In general, the global risk minimization in (4) can be extended for personalized FA as

$$\min_{\boldsymbol{\theta},\boldsymbol{\Phi}} \sum_{m\in\mathcal{M}} p_m \mathcal{R}_m(\boldsymbol{\theta}, \boldsymbol{\phi}_m; \mathcal{D}_m) \tag{5}$$

where $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_m\}_{m\in\mathcal{M}}$ collects personalized parameters $\boldsymbol{\phi}_m$ for all units $m \in \mathcal{M}$.

Methodologically, existing methods for personalized FA can be classified into two categories. One category includes two-step approaches which first (i) estimate a global model, say $\hat{\boldsymbol{\theta}}$, through collaborative learning across units (e.g., `FedAvg`), and then (ii) fine-tune $\boldsymbol{\phi}_m$ using local data $\mathcal{D}_m$ to find a personalized model within the neighborhood of $\hat{\boldsymbol{\theta}}$. Many personalization approaches fall into this category (e.g., Dinh, Tran, and Nguyen 2020; Hanzely and Richtárik 2020; Li et al. 2021; Shi and Kontar 2022) and some recent literature handles those two steps iteratively. The other category mainly focuses on DNNs, (e.g., Arivazhagan et al. 2019; Wang et al. 2019; Liang et al. 2020) where layers split into personalized and shared ones. Collaborative learning is performed on the weights of the shared layers while each unit estimates its personalized layer individually.

2.3. Challenges

Given the current FA literature, it is rather clear that employing an MGP (2) in IoFT poses significant challenges to parameter estimation and prediction. MGPs are built upon correlations and do not directly fit within an ERM paradigm. Specifically, the log-likelihood of the multivariate Gaussian in which between-output covariances $\text{cov}(f_m(\mathbf{x}), f_{m'}(\mathbf{x}'))$ for $m \neq m'$ should be calculated, which is infeasible without data sharing. Also, any local update to be done, needs to assess correlations with other clients. Thus, maximum likelihood estimation (MLE) for MGP needs to be rethought for FA. In addition, predictions of an MGP (3) require a posterior of latent variables given all data across units $p(\mathbf{f}|\mathbf{X}, \mathbf{y}; \boldsymbol{\theta}_{\text{full}}) = p(\mathbf{f}|\mathcal{D}_1, \ldots, \mathcal{D}_M; \boldsymbol{\theta}_{\text{full}})$. Unfortunately, this is not straightforward in federated settings where clients do not see each other's data.

In the following section, we will provide a simple yet effective solution to this challenge by taking a hierarchical modeling approach and remodeling dependencies to become amenable to federated inference.

3. Proposed Model

In this section, we introduce `FedMGP`, our proposed FA framework for MGP. In Section 3.1, we present model construction that is inspired by hierarchical Bayes. In Section 3.2, we develop an inference framework that exploits local computations to distribute computations and circumvent the need for data sharing. Finally, Section 3.3, discusses individualized predictions for each unit. Throughout the article, the superscript $^{\text{cov}}$ is placed for the hyperparameters of covariance functions.

3.1. Model Development

Our modeling strategy exploits the "hub-and-spoke" structure in Figure 1(b), a natural hierarchy exhibited in IoFT systems where one central server is connected to units at the edge. Our framework adopts this natural hierarchy by assuming that outputs of different units are generated from a set of common global latent functions. By collaboratively learning those latent functions, units can indirectly borrow strength from each other while circumventing the need to share raw data. Instead, only knowledge on the global functions needs to be shared.

Specifically, we model output functions $\{f_m(\mathbf{x})\}_{m\in\mathcal{M}}$ as GPs that depend on a common set of independent latent functions $\{g_i(\mathbf{x})\}_{i\in\mathcal{I}}$ indexed by $\mathcal{I} = \{1, \ldots, I\}$. Commonality across units is thus encoded and shared in the latent space. Figure 2 illustrates the correspondence between our approach and the natural hierarchy of the "hub-and-spoke" IoFT system. As shown in

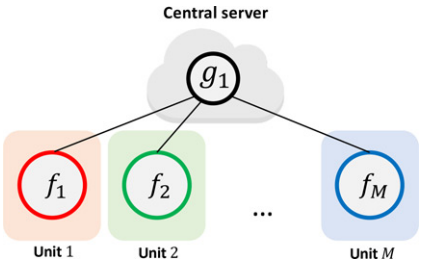

Figure 2. An illustration representing the correspondence between the IoFT system and the structure of the proposed MGP. A case with one latent function ($I = 1$) is presented.

Figure 2 our model is an instance of hierarchical Bayes (Koller and Friedman 2009) that treats data as realizations of a learnable latent probabilistic model ($\{f_m(\mathbf{x})\}_{m\in\mathcal{M}}$) parameterized by the stochastic processes ($\{g_i(\mathbf{x})\}_{i\in\mathcal{I}}$) at the higher hierarchical level. Under this hierarchical structure, units collaboratively infer $\{g_i(\mathbf{x})\}_{i\in\mathcal{I}}$, while, at the same time, each unit m estimates $f_m(\mathbf{x})$ using its local data $\mathcal{D}_m$. Therefore, updating the latent probabilistic model by learning local data yields an implicit update on the associated stochastic processes. In a federated regime, as the learning process happens at each unit using its local data, different local updates on the shared functions are produced from each unit. The central server thus collects and integrates local updates to make a global update on the shared functions; that is, the shared functions are learned collaboratively. Through this, knowledge is transferred across units.

A natural consequence of the hierarchical structure is that $f_1(\mathbf{x}), \ldots, f_M(\mathbf{x})$ are conditionally independent given $\{g_i(\mathbf{x})\}_{i\in\mathcal{I}}$. Now, let each $g_i(\mathbf{x})$ be a sample drawn from a GP with zero mean and covariance $c_{g_i,g_i}(\cdot,\cdot;\boldsymbol{\theta}_i^{\text{cov}})$. Then the probabilistic graphical model is written by

$$p\left(\{f_m(\mathbf{x})\}_{m\in\mathcal{M}}|\{g_i(\mathbf{x})\}_{i\in\mathcal{I}};\boldsymbol{\Phi}^{\text{cov}},\boldsymbol{\theta}^{\text{cov}}\right) = \prod_{m\in\mathcal{M}} p\left(f_m(\mathbf{x})|\{g_i(\mathbf{x})\}_{i\in\mathcal{I}};\boldsymbol{\phi}_m^{\text{cov}},\boldsymbol{\theta}^{\text{cov}}\right) \tag{6}$$

where $\boldsymbol{\theta}^{\text{cov}} = \{\boldsymbol{\theta}_i^{\text{cov}}\}_{i\in\mathcal{I}}$ and $\boldsymbol{\phi}_m^{\text{cov}} = \{\boldsymbol{\phi}_{m,i}^{\text{cov}}\}_{i\in\mathcal{I}}$ such that $\boldsymbol{\phi}_{m,i}^{\text{cov}}$ are hyperparameters simultaneously associated with $f_m(\mathbf{x})$ and $g_i(\mathbf{x})$ for $m \in \mathcal{M}$ and $i \in \mathcal{I}$. Here the hyperparameter notations (e.g., $\boldsymbol{\theta}^{\text{cov}}$ and $\boldsymbol{\phi}_m^{\text{cov}}$) emphasize the correspondence to the global and personalized parameters in the personalized FA framework (e.g., $\boldsymbol{\theta}$ and $\boldsymbol{\phi}_m$) discussed in Section 2.2.1.

Since shared latent processes $\{g_i(\mathbf{x})\}_{i\in\mathcal{I}}$ and conditional outputs $f_m(\mathbf{x})|\{g_i(\mathbf{x})\}_{i\in\mathcal{I}}$ are modeled as GPs, the marginal distribution for $\{f_m(\mathbf{x})\}_{m\in\mathcal{M}}$ can be expressed as a jointly distributed GP, that is, an MGP. Given (6), learning the MGP boils down to the estimation of the hyperparameters $\boldsymbol{\phi}_m^{\text{cov}}$ and $\boldsymbol{\theta}^{\text{cov}}$. Now the challenge is how to do this estimation in a federated setting where clients do not have access to each other's data, nor does the central server.

To this end, a starting point of our treatment is exploiting *pseudo-inputs* (also referred to as *inducing variables*), originally proposed for sparse approximation of GPs (Snelson and Ghahramani 2005). Pseudo-inputs are a set of latent input variables, which we do not observe, expected to well characterize a GP when output variables at their locations, called *pseudo-targets*, are evaluated. The key assumption in our federated setting is that the independence in (6) still holds if the latent GPs $\{g_i(\mathbf{x})\}_{i\in\mathcal{I}}$ are well characterized by some pseudo-inputs and their corresponding outputs. This assumption is a stepping stone for FA because we can evaluate $\{g_i(\mathbf{x})\}_{i\in\mathcal{I}}$ with the pseudo-inputs without knowing $\mathbf{X}_1, \ldots, \mathbf{X}_M$, which requires sharing $\{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$.

Consider J pseudo-inputs denoted as $\mathbf{W} = [\mathbf{w}_j^\top]_{j=1,\ldots,J}^\top$ and the corresponding outputs of the ith latent process $\mathbf{g}_i = [g_i(\mathbf{w}_j)]_{j=1,\ldots,J}^\top$. Also, let $\mathbf{g} = [\mathbf{g}_i^\top]_{i=1,\ldots,I}^\top$. The conditional independence in (6) with the pseudo-inputs can be written as

$$p(\mathbf{f}|\mathbf{g},\mathbf{X},\mathbf{W};\boldsymbol{\Phi}^{\text{cov}},\boldsymbol{\theta}^{\text{cov}}) = \prod_{m\in\mathcal{M}} p(\mathbf{f}_m|\mathbf{g},\mathbf{X},\mathbf{W};\boldsymbol{\phi}_m^{\text{cov}},\boldsymbol{\theta}^{\text{cov}}) = \prod_{m\in\mathcal{M}} \psi(\mathbf{f}_m;\mathbf{V}_m\mathbf{g},\boldsymbol{\Omega}_m), \tag{7}$$

$$p(\mathbf{g}|\mathbf{W};\boldsymbol{\theta}^{\text{cov}}) = \psi(\mathbf{g};\mathbf{0},\mathbf{C}_{\mathbf{g},\mathbf{g}}) \tag{8}$$

with

$$\mathbf{V}_m = \mathbf{C}_{\mathbf{f}_m,\mathbf{g}}\mathbf{C}_{\mathbf{g},\mathbf{g}}^{-1} \quad \text{and} \quad \boldsymbol{\Omega}_m = \mathbf{C}_{\mathbf{f}_m,\mathbf{f}_m} - \mathbf{V}_m\mathbf{C}_{\mathbf{g},\mathbf{f}_m}$$

such that $\mathbf{C}_{\mathbf{f}_m,\mathbf{g}} = \mathbf{C}_{\mathbf{g},\mathbf{f}_m}^\top = [\mathbf{C}_{\mathbf{f}_m,\mathbf{g}_i}]_{i=1}^I \in \mathbb{R}^{N_m\times IJ}$ denotes a matrix that concatenates cross-covariance matrices $\mathbf{C}_{\mathbf{f}_m,\mathbf{g}_i} \in \mathbb{R}^{N_m\times J}$ with $c_{f_m,g_i}(\cdot,\cdot;\boldsymbol{\phi}_{m,i}^{\text{cov}})$; $\mathbf{C}_{\mathbf{g},\mathbf{g}} = \text{bdiag}(\mathbf{C}_{\mathbf{g}_i,\mathbf{g}_i})_{i=1}^I \in \mathbb{R}^{IJ\times IJ}$ where $\mathbf{C}_{\mathbf{g}_i,\mathbf{g}_i} \in \mathbb{R}^{J\times J}$ is a covariance matrix with $c_{g_i,g_i}(\cdot,\cdot;\boldsymbol{\theta}_i^{\text{cov}})$; and $\mathbf{C}_{\mathbf{f}_m,\mathbf{f}_m}$ is built from $c_{f_m,f_m}(\cdot,\cdot;\boldsymbol{\phi}_m^{\text{cov}},\boldsymbol{\theta}^{\text{cov}})$. The hierarchical structure is summarized in Table 1.

Under the hierarchical GPs (7) and (8), elegant ways to design valid covariance functions c_{f_m,f_m} and c_{f_m,g_i} include the *linear model of coregionalization* (LMC) (Journel and Huijbregts 1976; Barry, Jay, and Hoef 1996) and *convolution processes* (CP) (Ver Hoef and Barry 1998; Higdon 2002; Boyle and Frean 2004). We will use CP-based covariance modeling in our experiments in Sections 5 and Section 6. The reader is referred to Section A of the supplementary material for details.

Given (7) and (8), we can see that cross-covariance matrices $\mathbf{C}_{\mathbf{f}_m,\mathbf{f}_{m'}}$ for $m \neq m'$ that requires $c_{f_m,f_m'} = \text{cov}(f_m(\mathbf{x}), f_{m'}(\mathbf{x}'))$ do not appear in the MGP built through our hierarchical structure. In other words, the cross-correlations between units can be characterized using $\mathbf{C}_{\mathbf{f}_m,\mathbf{f}_m}$, $\mathbf{C}_{\mathbf{f}_m,\mathbf{g}}$, and $\mathbf{C}_{\mathbf{g},\mathbf{g}}$, without the explicit calculation of cross-covariances $\mathbf{C}_{\mathbf{f}_m,\mathbf{f}_{m'}}$ which is not possible in a federated setting. Nonetheless, this by no means indicates that a traditional MLE framework can infer the MGP in a federated

Table 1. Hierarchical model.

Level	Variables	Model
1	Data	$[\mathbf{y}|\mathbf{f};\boldsymbol{\sigma}] = [y_{1,1}|f_{1,1};\sigma_1]\cdots[y_{M,N_M}|f_{M,N_M};\sigma_M]$
2	Output function	$[\mathbf{f}|\mathbf{g};\boldsymbol{\Phi}^{\text{cov}},\boldsymbol{\theta}^{\text{cov}}] = [\mathbf{f}_1|\mathbf{g};\boldsymbol{\phi}_1^{\text{cov}},\boldsymbol{\theta}^{\text{cov}}]\cdots[\mathbf{f}_M|\mathbf{g};\boldsymbol{\phi}_M^{\text{cov}},\boldsymbol{\theta}^{\text{cov}}]$
3	Shared latent function	$[\mathbf{g}|\boldsymbol{\theta}^{\text{cov}}] = [\mathbf{g}_1|\boldsymbol{\theta}_1^{\text{cov}}]\cdots[\mathbf{g}_I|\boldsymbol{\theta}_I^{\text{cov}}]$

setting. To see this, the marginal log-likelihood is

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{W}; \boldsymbol{\theta}^{\text{cov}}, \boldsymbol{\Phi}^{\text{cov}}, \boldsymbol{\sigma}) = \log \int p(\mathbf{y}|\mathbf{f}; \boldsymbol{\sigma}) p(\mathbf{f}|\mathbf{g}, \mathbf{X}, \mathbf{W}; \boldsymbol{\Phi}^{\text{cov}}) p(\mathbf{g}|\mathbf{W}; \boldsymbol{\theta}^{\text{cov}}) d\mathbf{f} d\mathbf{g}. \quad (9)$$

$$= \log \psi \left(\mathbf{y}; \mathbf{0}, \text{bdiag}(\boldsymbol{\Omega}_m)_{m=1}^M + \mathbf{V}\mathbf{C}_{\mathbf{g},\mathbf{f}} + \boldsymbol{\Sigma}\right), \quad (10)$$

with $\mathbf{V} = [\mathbf{V}_m^\top]_{m=1,\ldots,M}^\top$ and $\mathbf{C}_{\mathbf{g},\mathbf{f}} = [\mathbf{C}_{\mathbf{g},\mathbf{f}_m}]_{m=1}^M$. Directly maximizing (10) is still not possible as the objectives cannot be separated into marginal likelihoods of individual units. As such, a gradient calculation still requires all data at once. In the following section, we will see our approach to overcoming this issue, which builds upon our hierarchical construction and pseudo-inputs.

3.2. Federated Inference

Now we propose a federated inference procedure using VI. VI is an inference framework that finds a variational distribution approximating a posterior distribution. VI poses posterior inference as an optimization problem that minimizes the Kullback–Leibler (KL) divergence between the variational distribution and an often intractable posterior. This results in an *evidence lower bound* (ELBO) amenable to optimization algorithms. Indeed, recent studies have shown advantages of using VI in GPs (e.g., Laínez-Aguirre et al. 2016), including improved computational efficiency over alternatives (e.g., Salimans, Kingma, and Welling 2015) or intrinsic regularization properties of VI (e.g., Titsias 2009) that may lead to improved generalization (e.g., Yue and Al Kontar 2021).

Besides the above general advantages, we will show how we use a VI technique to naturally facilitate distributed learning for our hierarchical model characterized by pseudo-inputs. The key idea is to place a variational distribution on the shared latent functions and derive a decomposable ELBO over units. The decomposability allows units to learn the variational distribution collaboratively within a personalized FA framework.

Now let us see the details. Henceforth, we will omit hyperparameters (e.g., $\boldsymbol{\theta}^{\text{cov}}, \boldsymbol{\Phi}^{\text{cov}}, \boldsymbol{\sigma}$) unless there is ambiguity. Under VI, our goal is to approximate a posterior over the latent variables $\mathbf{f}$ and $\mathbf{g}$. To do so, we define a variational distribution over the global latent variables $q(\mathbf{g})$. As such, the joint distribution $q(\mathbf{f}, \mathbf{g})$ is given as

$$q(\mathbf{f}, \mathbf{g}) = p(\mathbf{f}|\mathbf{g}, \mathbf{X}, \mathbf{W}) q(\mathbf{g}) = \prod_{m \in \mathcal{M}} p(\mathbf{f}_m|\mathbf{g}, \mathbf{X}, \mathbf{W}) \prod_{i \in \mathcal{I}} q(\mathbf{g}_i) \quad (11)$$

with $q(\mathbf{g}) = \mathcal{N}(\boldsymbol{\mu}_\mathbf{g}, \mathbf{M}_{\mathbf{g},\mathbf{g}})$[1] encompassing independent Gaussian distributions $q(\mathbf{g}_i) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{g}_i}, \mathbf{M}_{\mathbf{g}_i,\mathbf{g}_i})$ such that $\boldsymbol{\mu}_\mathbf{g} = [\boldsymbol{\mu}_{\mathbf{g}_i}]_{i=1,\ldots,I}^\top$ and $\mathbf{M}_{\mathbf{g},\mathbf{g}} = \text{bdiag}(\mathbf{M}_{\mathbf{g}_i,\mathbf{g}_i})_{i=1}^I$. Notice here that (11) resembles a mean-field approximation (Blei, Kucukelbir, and McAuliffe 2017), yet in our case, is a natural consequence of our hierarchical structure.

[1] In practice, the matrix $\mathbf{M}_{\mathbf{g},\mathbf{g}}$ is reparameterized by a triangular matrix $\mathbf{L}_{\mathbf{g},\mathbf{g}}$ obtained through the Cholesky decomposition $\mathbf{M}_{\mathbf{g},\mathbf{g}} = \mathbf{L}_{\mathbf{g},\mathbf{g}}\mathbf{L}_{\mathbf{g},\mathbf{g}}^\top$ to guarantee positive semidefiniteness of $\mathbf{M}_{\mathbf{g},\mathbf{g}}$ throughout model training (Lindstrom and Bates 1988).

Our goal is to find the variational distribution $q(\mathbf{f}, \mathbf{g})$ to approximate the posterior $p(\mathbf{f}, \mathbf{g}|\mathcal{D}_1, \ldots, \mathcal{D}_M, \mathbf{W})$. The variational parameters $\boldsymbol{\mu}_\mathbf{g}$ and $\mathbf{M}_{\mathbf{g},\mathbf{g}}$ are universal across units $m \in \mathcal{M}$ as the associated variational distribution $q(\mathbf{g})$ approximates the posterior of shared latent GPs $p(\mathbf{g}|\mathcal{D}_1, \ldots, \mathcal{D}_M, \mathbf{W})$. Given (7) and (11), we can derive a variational marginal distribution $q(\mathbf{f}_m)$ from $q(\mathbf{f}, \mathbf{g})$ as

$$q(\mathbf{f}_m) = \int p(\mathbf{f}_m|\mathbf{g}, \mathbf{X}, \mathbf{W}) q(\mathbf{g}) d\mathbf{g} = \psi \left(\mathbf{f}_m; \mathbf{V}_m \boldsymbol{\mu}_\mathbf{g}, \boldsymbol{\Omega}_m + \mathbf{V}_m \mathbf{M}_{\mathbf{g},\mathbf{g}} \mathbf{V}_m^\top\right). \quad (12)$$

Now by a simple Jensen's inequality on (9), and using the variational distributions (11) and (12), we can derive a lower bound of the marginal log-likelihood (9) as

$$\log p(\mathbf{y}|\mathbf{X}) = \log \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{g}, \mathbf{X}, \mathbf{W}) p(\mathbf{g}|\mathbf{W}) d\mathbf{f} d\mathbf{g} \geq \int q(\mathbf{f}, \mathbf{g}) \log \frac{p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{g}, \mathbf{X}, \mathbf{W}) p(\mathbf{g}|\mathbf{W})}{q(\mathbf{f}, \mathbf{g})} d\mathbf{f} d\mathbf{g}.$$

We define ELBO as the lower bound above. It can be further reorganized in terms of $q(\mathbf{f}_m)$:

$$\text{ELBO} = \sum_{m \in \mathcal{M}} \mathbb{E}_{q(\mathbf{f}_m)} \left[\log p(\mathbf{y}_m|\mathbf{f}_m)\right] - \sum_{i \in \mathcal{I}} D_{\text{KL}}(q(\mathbf{g}_i) \| p(\mathbf{g}_i)) \quad (13)$$

where the notation $D_{\text{KL}}(\cdot\|\cdot)$ indicates the KL divergence.

The critical advantage of the ELBO in (13) is that its first term is decomposed into client-independent expectations that can be evaluated at each unit separately. While the second term is shared knowledge as it is a function of global latent variables. This indeed is similar to the personalized FA formulation in (5) where the risk function at each client can be written as

$$\mathcal{R}_m(\boldsymbol{\phi}_m, \boldsymbol{\theta}; \mathcal{D}_m) := \underbrace{-\frac{1}{p_m} \mathbb{E}_{q(\mathbf{f}_m)}[\log p(\mathbf{y}_m|\mathbf{f}_m)]}_{\text{Local model fit } \{\boldsymbol{\phi}_m, \boldsymbol{\theta}\}} + \underbrace{\sum_{i \in \mathcal{I}} D_{\text{KL}}(q(\mathbf{g}_i) \| p(\mathbf{g}_i))}_{\substack{\text{Regularization via} \\ \text{shared knowledge with } \boldsymbol{\theta}}} \quad (14)$$

where the ELBO is a sum over weighted local risks, that is, $\text{ELBO} = \sum_{m \in \mathcal{M}} p_m(-\mathcal{R}_m(\boldsymbol{\phi}_m, \boldsymbol{\theta}; \mathcal{D}_m))$, corresponding to the objective function in personalized FA (5). Here the global and personalized parameters of `FedMGP` are as follows: $\boldsymbol{\theta} := \{\boldsymbol{\theta}^{\text{cov}}, \boldsymbol{\mu}_\mathbf{g}, \mathbf{M}_{\mathbf{g},\mathbf{g}}, \mathbf{W}\}$ and $\boldsymbol{\phi}_m := \{\boldsymbol{\phi}_m^{\text{cov}}, \sigma_m\}$.

One can directly observe that the local risk function provides a natural interplay between shared and local knowledge. The first term encourages the variational density of latent variables to place probability mass on parameters that best explain the local data. While, the second term is a regularizer based on shared global knowledge. This naturally suggests an FA algorithm in the spirit of personalized FA where global parameters are shared and integrated into the central server. As such, we propose a federated algorithm to maximize (13) summarized in Algorithm 2. Note that, we assume that any operation on a parameter set corresponds to an operation on the vectorized set.

A detailed description for Algorithm 2 is as follows:

1. The central server deploys the global parameter $\boldsymbol{\theta}$ to all units $m \in \mathcal{M}$.

Algorithm 2 Inference of `FedMGP`

Input: $\boldsymbol{\theta}$ (the initial global parameters); $\{\boldsymbol{\phi}_m\}_{m\in\mathcal{M}}$ (the initial personalized parameters); η (the global step size); T (the total communication round); $\{E^{(t)}\}_{t=1}^{T}$ (local iterates at round t);

1: **for** the communication round $t = 1, \ldots, T$ **do**
2: The central server broadcasts $\boldsymbol{\theta}$ to all clients
3: **for** each client $m \in \mathcal{M}$ **do**
4: Update the local model:

$$\boldsymbol{\theta}_m^{\text{local}}, \boldsymbol{\phi}_m^{\text{local}} \leftarrow \texttt{client\_update}(\boldsymbol{\theta}, \boldsymbol{\phi}_m; \mathcal{D}_m, E^{(t)})$$

5: Update the personalized parameter: $\boldsymbol{\phi}_m \leftarrow \boldsymbol{\phi}_m^{\text{local}}$
6: Calculate a local pseudo-gradient: $\Delta_m^{\text{local}} \leftarrow \boldsymbol{\theta} - \boldsymbol{\theta}_m^{\text{local}}$
7: Upload Δ_m^{local} to the central server
8: **end for**
9: The central server calculates the global pseudo-gradient:

$$\Delta \leftarrow \texttt{cloud\_update}(\{\Delta_m^{\text{local}}\}_{m\in\mathcal{M}})$$

10: Update the global model: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\Delta$
11: **end for**

2. Each unit m updates $\boldsymbol{\theta}$ and $\boldsymbol{\phi}_m$ through `client_update`. Specifically, `client_update` minimizes $\mathcal{R}_m(\boldsymbol{\phi}_m, \boldsymbol{\theta}; \mathcal{D}_m)$ in (14) using an iterative algorithm (e.g., gradient-based methods). For example, one can take $E^{(t)}$ iterations of SGD to minimize $\mathcal{R}_m(\boldsymbol{\phi}_m, \boldsymbol{\theta}; \mathcal{D}_m)$ with respect to $\boldsymbol{\phi}_m$ and $\boldsymbol{\theta}$. After the iterates, `client_update` returns the updated parameters $\boldsymbol{\phi}_m^{\text{local}}$ and $\boldsymbol{\theta}_m^{\text{local}}$. The personalized parameter is then updated $\boldsymbol{\phi}_m \leftarrow \boldsymbol{\phi}_m^{\text{local}}$. A local pseudo-gradient is calculated as $\Delta_m^{\text{local}} \leftarrow \boldsymbol{\theta} - \boldsymbol{\theta}_m^{\text{local}}$ and sent back to the central server.
3. The central server aggregates the collected pseudo-gradients $\{\Delta_m^{\text{local}}\}_{m\in\mathcal{M}}$ by `cloud_update` to obtain the global pseudo-gradient Δ. Finally, the global parameter $\boldsymbol{\theta}$ is updated with a given step size η: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\Delta$.

The steps above are repeated until an exit condition is met. Throughout the proposed inferential algorithm, each unit's computing power is used when executing `client_update` and there is no raw data exchange. Through Algorithm 2, we can attain the estimates of original parameters and variational parameters. This is achieved in a communication-efficient way that allows units to take several local steps, a major advantageous feature for IoFT systems. Using the hat notation for estimates (e.g., $\hat{\boldsymbol{\theta}}$), we write the approximated posterior for $\mathbf{g}$ as $q(\mathbf{g}) = \mathcal{N}(\hat{\boldsymbol{\mu}}_{\mathbf{g}}, \hat{\mathbf{M}}_{\mathbf{g},\mathbf{g}}) \approx p(\mathbf{g}|\mathcal{D}_1, \ldots, \mathcal{D}_M, \mathbf{W})$, through which $q(\mathbf{f}_m) \approx p(\mathbf{f}_m|\mathcal{D}_1, \ldots, \mathcal{D}_M)$ is readily derived by (12).

3.2.1. *Aggregation Schemes in* `cloud_update`

In `cloud_update`, the global pseudo-gradient Δ is obtained by aggregating the received pseudo-gradients Δ_m^{local}. Like `FedAvg`, a simple way to do that is weighted averaging:

$$\Delta \leftarrow \sum_{m\in\mathcal{M}} p_m \Delta_m^{\text{local}}. \tag{15}$$

Indeed, recent work has provided elegant theoretical results on the convergence of distributed averaging schemes (e.g., `FedAvg`) in GPs and their advantageous properties (Yue and Kontar 2021). That being said, the proposed Algorithm 2 is a general framework that any aggregation scheme can be plugged into. We provide additional aggregation schemes in Section D and their empirical validation in Section E in the supplementary material. These include aggregation strategies for units with unstable communication or anomalous data.

3.2.2. *Stochastic Optimization in* `client_update`

One advantage of our VI construction is that it allows stochastic optimization methods such as SGD or Adam (Kingma and Ba 2015) to be used in `client_update`. This is attributed to the fact that the first term in (14) can be represented as the sum of independent terms over each individual observation: $\frac{1}{p_m}\mathbb{E}_{q(\mathbf{f}_m)}[\log p(\mathbf{y}_m|\mathbf{f}_m)] = \frac{1}{p_m}\mathbb{E}_{q(\mathbf{f}_m)}[\log\prod_{n=1}^{N_m} p(y_{m,n}|f_{m,n})] = \frac{1}{p_m}\sum_{n=1}^{N_m}\mathbb{E}_{q(f_{m,n})}[\log p(y_{m,n}|f_{m,n})] = \frac{1}{2p_m}\sum_{n=1}^{N_m}(-\log 2\pi - \log\sigma_m^2 - (y_{m,n}^2 + \mu_{m,n}^2 + \zeta_{m,n} - 2y_{m,n}\mu_{m,n})/\sigma_m^2)$ where $\mu_{m,n}$ is the nth element of the mean of $q(\mathbf{f}_m)$ and $\zeta_{m,n}$ is the nth element on the diagonal of the covariance matrix of $q(\mathbf{f}_m)$ in (12).

Through this decomposition, we can simply sample a batch and obtain its corresponding stochastic gradient. We note the idea of using stochastic optimization in GPs has been explored in an effort to handle an extremely large dataset (e.g., Hensman, Fusi, and Lawrence 2013; Nguyen et al. 2014; Moreno-Muñoz, Artés, and Álvarez 2018; Chung, Al Kontar, and Wu 2022; Chen et al. 2020). The focus of this literature however is on a GP under centralized regimes.

3.3. Personalized Prediction

Once units collaboratively infer the approximated posterior $q(\mathbf{g}) = \psi(\mathbf{g}; \hat{\boldsymbol{\mu}}_{\mathbf{g}}, \hat{\mathbf{M}}_{\mathbf{g},\mathbf{g}})$, each unit can derive its own posterior predictive distribution using $q(\mathbf{g})$ in place of the exact posterior $p(\mathbf{g}|\mathcal{D}_1, \ldots, \mathcal{D}_M, \mathbf{W})$. Let $\mathbf{X}_m^*$ denote new inputs at the unit m. A posterior predictive distribution of the corresponding output $\mathbf{y}_m^*$ is derived as

$$\begin{aligned} &p(\mathbf{y}_m^*|\mathbf{X}_m^*, \mathcal{D}_1, \ldots, \mathcal{D}_M, \mathbf{W}) \qquad (16)\\ &= \int p(\mathbf{y}_m^*|\mathbf{f}_m^*)p(\mathbf{f}_m^*|\mathbf{g}, \mathbf{X}_m^*, \mathbf{W})p(\mathbf{g}|\mathcal{D}_1, \ldots, \mathcal{D}_M, \mathbf{W})d\mathbf{f}_m^* d\mathbf{g}\\ &\approx \int p(\mathbf{y}_m^*|\mathbf{f}_m^*)p(\mathbf{f}_m^*|\mathbf{g}, \mathbf{X}_m^*, \mathbf{W})q(\mathbf{g})d\mathbf{f}_m^* d\mathbf{g}\\ &= \psi\left(\mathbf{y}_m^*; \mathbf{V}_m^*\hat{\boldsymbol{\mu}}_{\mathbf{g}}, \boldsymbol{\Omega}_m^* + \mathbf{V}_m^*\hat{\mathbf{M}}_{\mathbf{g},\mathbf{g}}\mathbf{V}_m^{*\top} + \hat{\sigma}_m^2\mathbf{I}\right) \end{aligned}$$

where all matrices are constructed with estimated parameters $(\hat{\boldsymbol{\phi}}_m, \hat{\boldsymbol{\theta}})$; the matrices $\mathbf{V}^*$ and $\boldsymbol{\Omega}_m^*$ are obtained in a similar fashion to $\mathbf{V}$ and $\boldsymbol{\Omega}$ but evaluated at $\mathbf{X}^*$. The ability of the proposed approach to quantify predictive uncertainty is a direct consequence of (16).

Remark 1. It is crucial to note that the calculation of the predictive distribution (16) involves the estimated local parameter $\hat{\boldsymbol{\phi}}_m$, and hence it is personalized to the local unit m. Also, deriving a predictive distribution is done locally at respective units; it needs neither other units' data nor computation at the central server. Yet, knowledge across all datasets is exploited via $q(\mathbf{g})$ which approximates the posterior $p(\mathbf{g}|\mathcal{D}_1, \ldots, \mathcal{D}_M, \mathbf{W})$.

4. Related Work

In an effort to mitigate the high computational cost associated with learning GPs, researchers have proposed various approaches that distribute the computational burden among multiple units with computing and data storage capabilities. These approaches can be classified into two categories: (i) module-based approaches and (ii) FA-based approaches.

In module-based methods, individual units train their own GP models independently until convergence, after which the models or their predictions are combined to form a global GP prediction. Some methods are based on the mixture of experts framework (Tresp 2000; Cao and Fleet 2014; Deisenroth and Ng 2015), which assumes that local GPs are independent to allow the global marginal likelihood to be decomposed into local marginal likelihoods. While these methods require only one round of communication, they are prone to issues of variance over-estimation (Cao and Fleet 2014) or weak experts (Tresp 2000) due to the factorization. To address these issues, Deisenroth and Ng (2015) introduce additional hyperparameters that reflect each expert's contribution, yet these are determined in a heuristic manner.

Compared to combining summary information at the end, the FA framework is a collaborative process that operates under multiple communication rounds. Such a collaborative process is shown to be advantageous both theoretically and empirically compared to combining summary information after training locally till convergence. For example, Li et al. (2019) show that the convergence bound of `FedAvg` follows $\mathcal{O}(E/T)$ with local steps E and the total communication rounds T, implying that E should be small enough for the convergence of `FedAvg`. Inspired by the FA framework, there has been recent interest in approaches to learning GPs in a federated way. Kontoudis and Stilwell (2022) investigate an alternating direction method of multipliers-based method for decentralized GP learning in situations where none of or a part of local datasets can be shared across adjunct units. This approach assumes the independence of local GPs and, thus, the decomposition of the global marginal likelihood. Achituve et al. (2021) present an FA framework for personalized classification, which involves learning a global deep kernel along with personalized GP classifiers. Yu et al. (2022) use a two-stage federated method to learn a global GP with a deep random kernel. Yue and Kontar (2021) investigate the convergence of `FedAvg` when applied directly to learning GP hyperparameters and highlight the associated empirical benefits. Note that these methods are all designed for single-output GPs. While our proposed `FedMGP` falls into the category of FA-based approaches, its uniqueness is that (i) it learns an MGP where a key additional challenge is the need to learn the between-unit covariances and (ii) it employs a variational learning approach to enable an FA framework for MGPs.

In the context of variational learning of a GP using locally stored data and local computing resources, `ModularGP` (Moreno-Muñoz, Artes, and Álvarez 2021) is a module-based approach that aims to discover a global MGP that can learn from all local datasets without direct access to them. Instead, it uses a variational predictive distribution from GP modules trained on local datasets. More specifically, each unit trains a stochastic variational GP (Hensman, Fusi, and Lawrence 2013) independently using its local dataset. Upon completion of the local learning process, units transmit dictionaries to the central server, consisting of GP hyperparameters, variational distributions of pseudo-targets, and the value of the maximized variational lower bound, but not the local datasets. The central server then builds a lower bound based on the received dictionaries, and maximizes the lower bound to infer a global variational distribution that approximates the posterior of the global MGP given all local data. Notice that, `ModularGP` estimates an MGP such that its predictive distribution for unit m is close to the predictive posterior distribution of the pseudo-targets received from unit m. In contrast, our `FedMGP` directly fits the predictive distribution to the likelihood, enabling direct use of the data rather than relying on estimated pseudo-targets. Moreover, by sharing locally learned variational distributions, `ModularGP` exposes the underlying data and trends of each local unit, posing significant privacy risks. On the other hand, `FedMGP` only shares common latent functions amongst units while keeping personalized parameters stored locally. Consequently, `FedMGP` can enhance privacy while achieving the desired objectives. Section B in the supplementary material provides an in-depth discussion that highlights the key differences between `FedMGP` and `ModularGP`.

5. Simulation Study

In this section, we evaluate our proposed `FedMGP` using simulated data. We consider two scenarios: nonparametric signal extrapolation (Section 5.1), IoFT systems with different scales (Section 5.2). Note that, in the supplementary material, we also discuss additional scenarios with units that have unstable communication (Section E.1) and with the participation of anomalous units producing highly heterogeneous signals (Section E.2).

We examine a `FedMGP`-based model and benchmark models as follows:

- `FedMGP-avg`: our proposed `FedMGP` that aggregates local pseudo-gradients through averaging; shown in (15).
- `IndGP`: an independent single GP deployed to each unit.
- `CenMGP`: a centralized MGP built on (7) and (8). While it is expected that `CenMGP` performs best among considered models, comparing with `CenMGP` will shed light on the predictive competitiveness of `FedMGP`, our federated approach.
- `ModularGP`: a module-based approach to learn an MGP (Moreno-Muñoz, Artes, and Álvarez 2021).
- `FedPoly`: a federated polynomial regression with personalization. This is a simple two-step personalized FA approach where (i) a polynomial regression model in the form of $f^{\text{poly}}(x;\boldsymbol{\theta}) = \sum_{k=0}^{K}\theta_k x^k$ whose parameter $\boldsymbol{\theta} := [\theta_k]_{k=0}^{K}$ is trained by `FedAvg` across units, and then (ii) personalized to the local data $\mathcal{D}_m$ by minimizing a penalized least squares loss function: $\min_{\boldsymbol{\theta}} \frac{1}{N_m}\sum_{n=1}^{N_m}\ell(f^{\text{poly}}(\mathbf{x}_{m,n};\boldsymbol{\theta}), y_{m,n}) - \omega\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|_2^2$ where $\boldsymbol{\theta}^*$ is the estimated parameter at the first step and ω is a positive coefficient. Therefore, this approach personalizes $\boldsymbol{\theta}$ to each unit $m \in \mathcal{M}$ while retaining global knowledge by encouraging a solution close to $\boldsymbol{\theta}^*$ estimated across units.

For IndGP, covariances are calculated by the radial basis function (RBF) kernel. For all MGPs, the latent GP $g_i(\mathbf{w})$ is constructed with a covariance function defined by

$$c_{g_i,g_i}(\mathbf{w},\mathbf{w}';\boldsymbol{\theta}_i^{\text{cov}}) := \exp\left(-\frac{1}{2}(\mathbf{w}-\mathbf{w}')^\top \mathbf{S}_i^{-1}(\mathbf{w}-\mathbf{w}')\right) \quad (17)$$

where $\mathbf{S}_i \in \mathbb{R}_+^{d\times d}$ is a diagonal matrix for length scale parameters. Therefore, $\boldsymbol{\theta}_i^{\text{cov}} \equiv \mathbf{S}_i$. The cross-covariances between $g_i(\mathbf{w})$ and $f_m(\mathbf{x})$ is defined as

$$c_{f_m,g_i}(\mathbf{x},\mathbf{w};\boldsymbol{\phi}_{m,i}^{\text{cov}},\boldsymbol{\theta}_i^{\text{cov}}) := \sqrt{\frac{v_{m,i}^2 \det(\mathbf{S}_i)}{\det(\mathbf{R}_{m,i}+\mathbf{S}_i)}}\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{w})^\top(\mathbf{R}_{m,i}+\mathbf{S}_i)^{-1}(\mathbf{x}-\mathbf{w})\right) \quad (18)$$

where $\boldsymbol{\phi}_{m,i}^{\text{cov}} \equiv \{\mathbf{R}_{m,i}, v_{m,i}\}$ defines unit-specific parameters. Indeed, (18) is a widely used covariance function (e.g., Majumdar and Gelfand 2007; Álvarez and Lawrence 2011; Fricker, Oakley, and Urban 2013; Kontar et al. 2018; Li et al. 2018; Chung, Al Kontar, and Wu 2022) for non-separable MGPs where $f_m(\mathbf{x})$ is defined as a convolution (Higdon 2002) of a Gaussian smoothing kernel with a shared latent white noise process $g_i(\mathbf{w})$. In all simulation studies we use one latent function ($I = 1$) for FedMGP, ModularGP, and CenMGP.

Finally, model evaluation is based on 30 experiment replications. We refer the reader to Section C in the supplementary material for details of implementation settings.

5.1. Nonparametric Signal Extrapolation

Our task in this simulation is to extrapolate signal observations from a target unit, by borrowing information from other units. Specifically, the target unit has a signal truncated at some point, whereas other units have the entire length of their signals. The simulation implicates the potential usage of FedMGP for predicting the future evolution of condition monitoring signals collected from operating engineering components or recovering a missing part of a signal possibly due to sudden sensor malfunctions. This simulation compares FedMGP-avg with FedPoly, ModularGP, IndGP, and CenMGP.

5.1.1. Setup

To generate data, we first obtain a smooth curve $\tilde{g}(\cdot)$ drawn from a GP with the RBF kernel with a length scale of 0.1 in a $d = 1$ dimensional space. The data generating model for unit m is constructed using a convolution:

$$y = \delta_m \int_{-\infty}^{\infty} h_m(x-u)\tilde{g}(u)du + \epsilon_m\,, \quad (19)$$

where $h_m(\cdot) = \psi(\cdot;0,\lambda_m)$ is a one-dimensional Gaussian smoothing kernel, $\delta_m \in \mathbb{R}$ is a parameter that controls signal amplitude and $\epsilon_m \sim \mathcal{N}(0,\sigma_m^2)$. Here, units share common latent functions $\tilde{g}(\cdot)$, yet convolved with different smoothing kernels $h_m(\cdot)$ and parameters δ_m, to allow both shared and unique features.

Next, we generate observations from $M = 5$ units. For each unit, we generate 200 evenly-spaced $d = 1$ dimensional input points ranging over $[-1,1]$. We then randomly draw five iid samples of $\delta_m \sim \text{unif}(0.5,3)$ and $\lambda_m \sim \text{unif}(2,10)$, where unif stands for the uniform distribution. σ_m is set to 0.1. Using (19) and the sampled parameters, we obtain 200 output points for each unit evaluated at the input points. Unit 1 is the target unit where observations belonging in $(0,1]$ are removed. The mean squared error (MSE) for each unit is assessed for the missing range.

For FedMGP-avg, ModularGP, and CenMGP, we place $J = 30$ pseudo-inputs evenly spaced within $[-1.1,1.1]$. We also place 30 pseudo-inputs for each GP module in the independent learning stage of ModularGP. To further see the robustness of FedMGP-avg in the selection of pseudo-inputs, we examine FedMGP-avg performance when pseudo-inputs are perturbed by Gaussian random variables with zero mean and variances 0.05^2 and 0.1^2.

5.1.2. Results

Results are presented in Table 2 and Figure 3. Many insights can be obtained from the results. Based on the average MSEs in Table 2, we observe that FedMGP-avg significantly outperforms IndGP. This demonstrates that our approach allows units to collaboratively learn an integrative model that results in better performance compared to units using their own data only. Figure 3 clearly shows such advantages, where FedMGP-avg accurately extrapolates the target unit's signal in $[0,1]$ with low predictive variance, whereas IndGP fails to do that. Second, FedMGP-avg achieves comparable performance to CenMGP. This highlights the ability to avoid raw data sharing and reduce computing/storage needs at the central server without sacrificing predictive accuracy. Third, as presented in Figure 3, signals exhibit different yet related trends. Our approach can learn a personalized model accounting for the unit-specific features. Indeed, personalization is a challenge in traditional FA built upon DNNs. The FedMGP-avg's superior personalization ability is further highlighted through improved performance over FedPoly. The improved personalization is due to the intrinsic advantage of MGP construction, which introduces both personalized and global parameters $(\boldsymbol{\phi}_m, \boldsymbol{\theta})$. Fourth, perturbing pseudo-inputs does not significantly affect the performance of FedMGP-avg, demonstrating the robustness of FedMGP-avg in choosing pseudo-inputs. Fifth, FedMGP-avg provides better predictive accuracy than ModularGP. It supports what we mentioned in Section 4, that ModularGP could deteriorate performance because it fits an MGP to the approximate posterior distributions of pseudo-targets received from GP modules rather than data likelihoods. Finally, our approach can quantify predictive uncertainty quite well. In the upper-left panel of Figure 3, we visually see that the 99% prediction intervals of FedMGP-avg are well-quantified even in $(0,1]$ for unit 1 where observations are missing. This is expected, as extrapolation for unit 1 can be seen as an interpolation across units.

5.2. IoFT System with Different Scales

This simulation examines our approach in IoFT systems with different scales. Cases with a moderate or large number of units M and observations N_m are considered. The simulation explores in what situations our proposed approach can be more effective

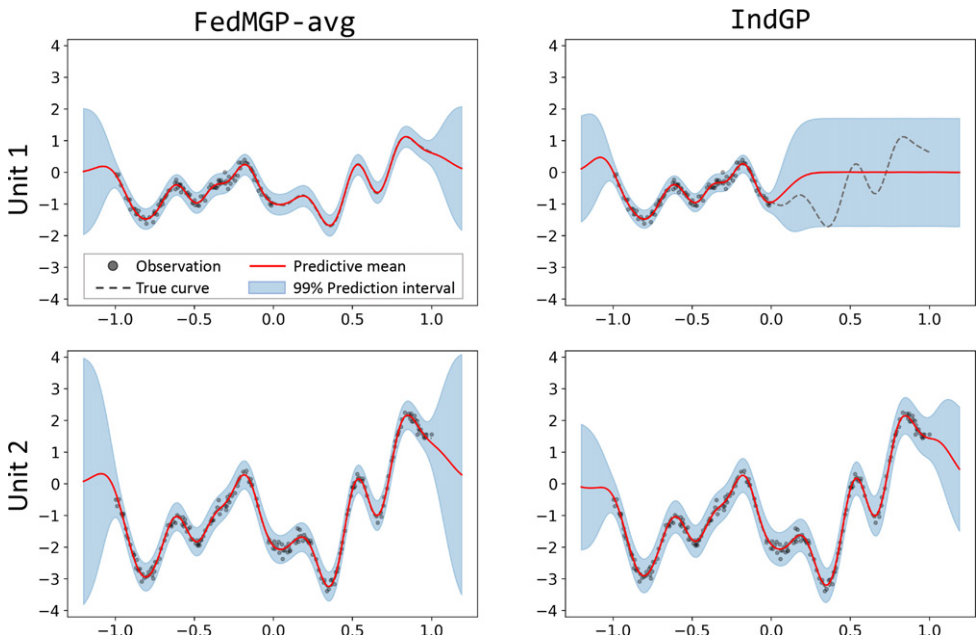

Figure 3. Nonparametric signal extrapolation.

Table 2. Comparison of MSEs evaluated at the missing range.

	FedMGP-avg	FedMGP-avg (0.05^2)	FedMGP-avg (0.1^2)	FedPoly	ModularGP	IndGP	CenMGP
Avg.	**0.012**	0.013	**0.012**	0.057	0.114	2.226	0.010
Std.	0.003	0.004	0.002	0.19	0.109	2.112	0.002

NOTE: Values in parentheses indicate the variance of the Gaussian noise added to the pseudo-inputs of FedMGP-avg. Best results among models except for CenMGP are boldfaced.

Table 3. Average MSEs over units.

Model	(N_m, M)			
	(20, 10)	(20, 200)	(1000, 10)	(1000, 200)
FedMGP-avg	**0.0119** (0.001)	**0.0151** (0.010)	**0.0103** (0.000)	**0.0137** (0.008)
IndGP	0.0581 (0.028)	0.0678 (0.025)	0.0118 (0.003)	0.0174 (0.016)
CenMGP	0.0119 (0.001)	0.0132 (0.0)	0.0109 (0.001)	– (–)

NOTE: Values in parentheses indicate standard deviations over repeated experiments. Best results among models except for CenMGP are boldfaced. Note that CenMGP is inscalable to the case with $(N_m, M) = (1000, 200)$.

compared to independent or centralized models. We compare FedMGP-avg, IndGP, and CenMGP.

5.2.1. Setup

We use the same data generating model as (19), setting $\delta_m \sim \text{unif}(0.5, 3)$ and $\lambda_m \sim \text{unif}(8, 10)$. We run experiments under four cases where (N_m, M) is set to (20, 10), (20, 200), (1000, 10), and (1000, 200). That is, models are trained on 200k observations in the largest-scale case. For those with $N_m = 1000$, we used stochastic optimization for local training in FedMGP with batch size 10. Average MSE over units is used to evaluate model performance.

5.2.2. Results

Results in Table 3 show that FedMGP performs better than IndGP in all cases, while comparable to CenMGP. Specifically when local data is sparse (the cases with $N_m = 20$), FedMGP significantly outperforms IndGP. This is because observations of each unit are not enough for IndGP to infer the true curve independently. On the other hand, FedMGP can transfer knowledge across units, yet without sharing their data, resulting in quite accurate predictions. Moreover, results show the scalability of FedMGP that can distribute model learning efforts to the units by exploiting local computing power when a large number of units with many observations participate in the system, for example, $(N_m, M) = (1000, 200)$, where CenMGP fails to scale.

6. Case Study: Battery Degradation Signal Prediction

In this section, we present an application of our approach to data-driven predictive analytics for reliability engineering. Forecasting the future degradation trend plays an important role in reliability engineering. Today's connectivity across units, along with reduced data acquisition cost, facilitates data-driven forecasting approaches, where the degradation signal of an in-service unit can be modeled based on the data from other connected units. This case study considers data-driven modeling of battery degradation under the scenario where units operating

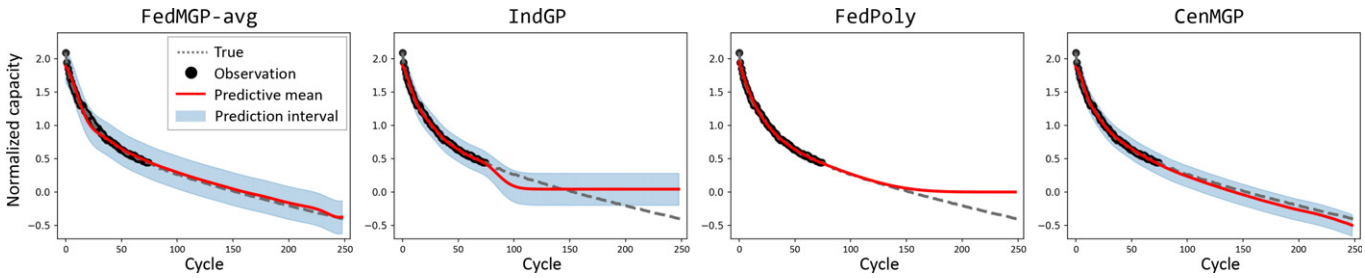

Figure 4. Predictions for Battery Cell 9's degradation signal observed up to 30%.

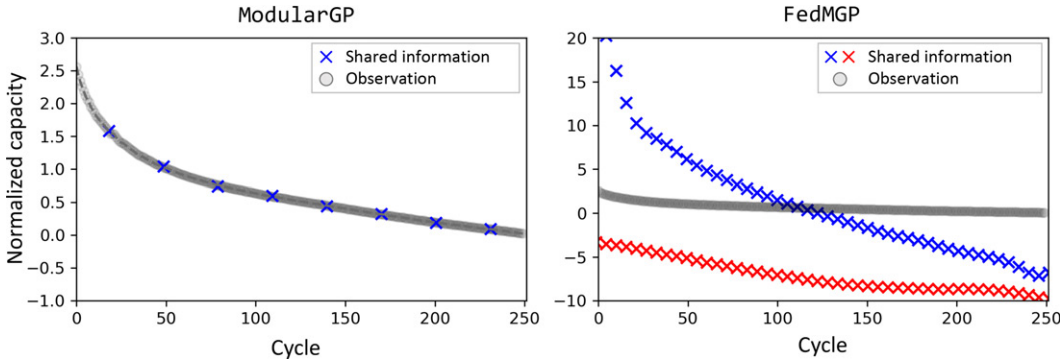

Figure 5. A trend of Battery Cell 5's capacity decrease and the shared information of `ModularGP` and `FedMGP-avg`. Note that the scale of y-axis is different.

Table 4. Average MSEs ($\times 10^{-3}$) in different degradation phases.

Model	30%	50%	70%
`FedMGP-avg`	**1.94** (1.03)	**1.68** (0.97)	**1.44** (1.06)
`FedPoly`	6.13 (2.11)	3.98 (1.38)	2.25 (1.77)
`ModularGP`	2.11 (1.49)	1.79 (1.35)	1.48 (1.39)
`IndGP`	9.63 (7.50)	3.62 (0.72)	4.17 (1.62)
`CenMGP`	1.88 (0.72)	1.65 (0.79)	1.41 (0.82)

NOTE: Values in parentheses indicate standard deviations of MSEs. Best results among models except for `CenMGP` are boldfaced.

individual batteries are also equipped with computing capabilities. We employ our proposed `FedMGP` to let the units use their respective computing resources to construct an MGP-based prognostics model in a collaborative manner. Thus, our model predicts battery degradation by leveraging other units' battery information and computing resources, yet without necessitating data sharing with a central server.

6.1. Setup

We use the CALCE battery team's dataset (Lee, Kwon, and Pecht 2018). The dataset comprises degradation signals from $M = 23$ batteries run over $N_m = 250$ cycles. A capacity (mAh) decrease in consumption cycles defines battery degradation given a nominal capacity 350 mAh. We randomly choose one test battery and remove a part of its signal to consider three cases where data is observed up to the 75th (30%), 125th (50%), and 175th (70%) cycle, respectively. We repeat the experiment for each case five times. We compare our proposed framework `FedMGP-avg` with four benchmark models `IndGP`, `FedPoly`, `ModularGP`, and `CenMGP`. In particular, polynomial regression in `FedPoly` fits the logarithm of the response variable, which is a widely used setting to characterize battery degradation signals in the literature (e.g., Son et al. 2013; Kontar et al. 2017). By cross-validation, we set the degree of polynomial regression in `FedPoly` to $K = 4$ and use $I = 2$ latent functions for `FedMGP-avg`, `ModularGP`, and `CenMGP`. Each latent function is evaluated at $J = 50$ evenly spaced pseudo-inputs. In the independent learning stage in `ModularGP`, we use 8 pseudo inputs evenly spaced.

6.2. Results

For each model, we provide the average of MSEs and their standard deviation over repeated experiments in Table 4. From the results, it is clear that `FedMGP-avg` achieves remarkably better performance for all degradation stages than `IndGP` and `FedPoly`, highlighting the practicality of our approach. A comparative illustration of predicted degradation curves is shown in Figure 4. The figure shows that independent modeling by `IndGP` and two-step personalized modeling by `FedPoly` fail to provide either adequate predictions or predictive uncertainty. In particular, it is not straightforward to quantify predictive uncertainty in two-step approaches such as `FedPoly`. On the contrary, `FedMGP-avg` provides accurate estimates as well as well-quantified predictive uncertainty. Indeed, uncertainty quantification can be extremely useful in determining optimal inspection and maintenance decisions in reliability engineering (Birolini 2013). Also, results over different degradation stages in Table 4 indicate that the prediction accuracy of `FedMGP-avg` improves as more data is obtained. Finally, `ModularGP` suffers not only deteriorated performance but also privacy risk. Figure 5 depicts the observations of the capacity of Battery Cell 5 (grey dots) and shared information (cross marks). Recall that `ModularGP` shares a pseudo-target distribution estimated by each unit, while what `FedMGP` shares are merely common latent functions $\{q(\mathbf{g}_i)\}_{i\in\{1,2\}}$. The figure pictorially demonstrates that

ModularGP can divulge battery's degradation trends by sharing the pseudo-target distribution, while FedMGP does not.

7. Conclusion and Discussion

The increase in computing power of edge devices in today's IoT provides an opportunity to distribute model learning efforts to reduce costs, achieve local decisions, and preserve privacy. As a result, many efforts have been devoted to the development of FA algorithms in the past few years, with the main focus on deep learning models. However, MGPs have yet to come into the spotlight of FA despite their natural application to integrative analysis of data from multiple IoT devices. Our study thus fills the gap between the current MGP and FA literature.

Inspired by the natural hierarchy of an IoFT system, we construct a hierarchical modeling and learning approach for an MGP built upon shared global latent functions. We then propose a VI approach that is amenable for distributed inference and overcomes the need to share raw data. Instead, only parameters of the shared global latent functions need to be shared to infer our model. Through comprehensive simulation and case studies, we present the advantageous properties of our model compared to both centralized and separate modeling approaches.

There are many potential existing directions to extend our model. One possible direction is to incorporate deep GPs (e.g., Damianou and Lawrence 2013). Deep GPs possess powerful representation capability and nonstationary flexibility, achieved by stacking multiple layers of GPs. Such merits allow its broad use in multi-fidelity modeling (e.g., Perdikaris et al. 2017) or computer simulation experiments (e.g., Sauer, Gramacy, and Higdon 2023). One approach is to place multiple GPs at the last layer that shares previous layers, yielding multiple correlated outputs (i.e., an MGP). Yet, this creates a fundamental challenge to estimating the complex model in a federated fashion. Like our approach, deriving a variational lower bound decomposable across units could be a possible approach. Another possible generalization of FedMGP is to build a model capable of handling data of various types or accounting for qualitative factors, which are common in IoFT practices. For example, vehicles in an IoFT-enabled telematics system may be equipped with sensors that collect data of different types (e.g., continuous or discrete variables) or may be grouped based on their qualitative features (e.g., sedans or trucks). Incorporating heterogeneous MGPs (Moreno-Muñoz, Artés, and Álvarez 2018) or kernels designed for both quantitative and qualitative factors (Deng et al. 2017) into our framework are promising future directions to pursue.

Supplementary Materials

Additional discussions and numerical results: The file (suppl.pdf) contains an additional discussion on the covariance construction of MGPs (Section A), an in-depth review of ModularGP and its comparison to FedMGP (Section B), detailed implementation setup (Section C), alternative aggregation schemes in cloud_update beyond simple averaging (Section D), and the corresponding additional experiments (Section E).

Code: The file (code.zip) contains codes that reproduce some of the results in the article.

Disclosure Statement

No potential conflict of interest was reported by the authors.

ORCID

Raed Al Kontar https://orcid.org/0000-0002-4546-324X

References

Achituve, I., Shamsian, A., Navon, A., Chechik, G., and Fetaya, E. (2021), "Personalized Federated Learning with Gaussian Processes," in *Advances in Neural Information Processing Systems* (Vol. 34), pp. 8392–8406, Curran Associates, Inc. [97]

Álvarez, M. A., and Lawrence, N. D. (2011), "Computationally Efficient Convolved Multiple Output Gaussian Processes," *Journal of Machine Learning Research*, 12, 1459–1500. [98]

Arivazhagan, M. G., Aggarwal, V., Singh, A. K., and Choudhary, S. (2019), "Federated Learning with Personalization Layers," arXiv preprint arXiv:1912.00818. [93]

Bae, B., Kim, H., Lim, H., Liu, Y., Han, L. D., and Freeze, P. B. (2018), "Missing Data Imputation for Traffic Flow Speed Using Spatio-Temporal Cokriging," *Transportation Research Part C: Emerging Technologies*, 88, 124–139. [90]

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. (2020), "How to Backdoor Federated Learning," in *International Conference on Artificial Intelligence and Statistics* (Vo. 108), pp. 2938–2948, PMLR. [93]

Barry, R. P., Jay, M., and Hoef, V. (1996), "Blackbox Kriging: Spatial Prediction Without Specifying Variogram Models," *Journal of Agricultural, Biological, and Environmental Statistics*, 1, 297–322. [94]

Birolini, A. (2013), *Reliability Engineering: Theory and Practice*, Berlin: Springer. [100]

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017), "Variational Inference: A Review for Statisticians," *Journal of the American statistical Association*, 112, 859–877. [95]

Boyle, P., and Frean, M. (2004), "Dependent Gaussian Processes," in *Advances in Neural Information Processing Systems* (Vol. 17), MIT Press. [94]

Cao, Y., and Fleet, D. J. (2014), "Generalized Product of Experts for Automatic and Principled Fusion of Gaussian Process Predictions," arXiv preprint arXiv:1410.7827. [97]

Chen, H., Zheng, L., AL Kontar, R., and Raskutti, G. (2020), "Stochastic Gradient Descent in Correlated Settings: A Study on Gaussian Processes," in *Advances in Neural Information Processing Systems* (Vol. 33), pp. 2722–2733, Curran Associates, Inc. [90,96]

Chen, J., Mak, S., Joseph, V. R., and Zhang, C. (2021), "Function-on-Function Kriging, with Applications to Three-Dimensional Printing of Aortic Tissues," *Technometrics*, 63, 384–395. [90]

Cheng, L.-F., Dumitrascu, B., Darnell, G., Chivers, C., Draugelis, M., Li, K., and Engelhardt, B. E. (2020), "Sparse Multi-Output Gaussian Processes for Online Medical Time Series Prediction," *BMC Medical Informatics and Decision Making*, 20, 1–23. [90]

Chung, S., Al Kontar, R., and Wu, Z. (2022), "Weakly Supervised Multi-Output Regression via Correlated Gaussian Processes," *INFORMS Journal on Data Science*, 1, 115–137. [90,96,98]

Damianou, A., and Lawrence, N. D. (2013), "Deep Gaussian Processes," in *Artificial Intelligence and Statistics* (Vol. 31), pp. 207–215, PMLR. [101]

Deisenroth, M., and Ng, J. W. (2015), "Distributed Gaussian Processes," in *International Conference on Machine Learning* (Vol. 37), pp. 1481–1490, PMLR. [97]

Deng, X., Lin, C. D., Liu, K.-W., and Rowe, R. (2017), "Additive Gaussian Process for Computer Models with Qualitative and Quantitative Factors," *Technometrics*, 59, 283–292. [101]

Fricker, T. E., Oakley, J. E., and Urban, N. M. (2013), "Multivariate Gaussian Process Emulators with Nonseparable Covariance Structures," *Technometrics*, 55, 47–56. [98]

Gotway, C. A., and Young, L. J. (2002), "Combining Incompatible Spatial Data," *Journal of the American Statistical Association*, 97, 632–648. [90]

Guhaniyogi, R., and Banerjee, S. (2018), "Meta-Kriging: Scalable Bayesian Modeling and Inference for Massive Spatial Datasets," *Technometrics*, 60, 430–444. [90]

Handcock, M. S., and Stein, M. L. (1993), "A Bayesian Analysis of Kriging," *Technometrics*, 35, 403–410. [90]

Hanzely, F., and Richtárik, P. (2020), "Federated Learning of a Mixture of Global and Local Models," arXiv preprint arXiv:2002.05516. [93]

Hensman, J., Fusi, N., and Lawrence, N. D. (2013), "Gaussian Processes for Big Data," in *Uncertainty in Artificial Intelligence*, pp. 282–290, Arlington, VI: AUAI Press. [96,97]

Higdon, D. (2002), "Space and Space-Time Modeling Using Process Convolutions," in *Quantitative Methods for Current Environmental Issues*, eds. C. W. Anderson, V. Barnett, P. C. Chatwin, and A. H. El-Shaarawi, pp. 37–56, London: Springer. [94,98]

Huang, J., and Gramacy, R. B. (2021), "Multi-Output Calibration of a Honeycomb Seal via On-site Surrogates," arXiv preprint arXiv:2102.00391. [90]

Journel, A. G., and Huijbregts, C. J. (1976), *Mining Geostatistics*, Caldwell, NJ: The Blackburn Press. [94]

Kingma, D. P., and Ba, J. (2015), "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*. [96]

Koller, D., and Friedman, N. (2009), *Probabilistic Graphical Models: Principles and Techniques*, Cambridge, MA: MIT Press. [94]

Kontar, R., Shi, N., Yue, X., Chung, S., Byon, E., Chowdhury, M., Jin, J., Kontar, W., Masoud, N., Nouiehed, M., et al. (2021), "The Internet of Federated Things (IoFT)," *IEEE Access*, 9, 156071–156113. [90,93]

Kontar, R., Son, J., Zhou, S., Sankavaram, C., Zhang, Y., and Du, X. (2017), "Remaining Useful Life Prediction based on the Mixed Effects Model with Mixture Prior Distribution," *IISE Transactions*, 49, 682–697. [100]

Kontar, R., Zhou, S., Sankavaram, C., Du, X., and Zhang, Y. (2018), "Nonparametric Modeling and Prognosis of Condition Monitoring Signals using Multivariate Gaussian Convolution Processes," *Technometrics*, 60, 484–496. [90,98]

Kontoudis, G. P., and Stilwell, D. J. (2022), "Fully Decentralized, Scalable Gaussian Processes for Multi-Agent Federated Learning," arXiv preprint arXiv:2203.02865. [97]

Laínez-Aguirre, J. M., Mockus, L., Orçun, S., Blau, G., and Reklaitis, G. V. (2016), "A Decomposition Strategy for the Variational Inference of Complex Systems," *Technometrics*, 58, 84–94. [95]

Lee, J., Kwon, D., and Pecht, M. G. (2018), "Reduction of Li-ion Battery Qualification Time based on Prognostics and Health Management," *IEEE Transactions on Industrial Electronics*, 66, 7310–7315. [100]

Li, J., and Zimmerman, D. L. (2015), "Model-based Sampling Design for Multivariate Geostatistics," *Technometrics*, 57, 75–86. [90]

Li, T., Hu, S., Beirami, A., and Smith, V. (2021), "Ditto: Fair and Robust Federated Learning through Personalization," in *International Conference on Machine Learning* (Vol. 139), pp. 6357–6368. PMLR. [93]

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020a), "Federated Optimization in Heterogeneous Networks," in *Proceedings of Machine Learning and Systems* (Vol. 2), pp. 429–450. [93]

Li, T., Sanjabi, M., Beirami, A., and Smith, V. (2020b), "Fair Resource Allocation in Federated Learning," in *International Conference on Learning Representations*. [93]

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. (2019), "On the Convergence of FedAvg on non-*i.i.d* Data," arXiv preprint arXiv:1907.02189. [97]

Li, Y., Zhou, Q., Huang, X., and Zeng, L. (2018), "Pairwise Estimation of Multivariate Gaussian Process Models with Replicated Observations: Application to Multivariate Profile Monitoring," *Technometrics*, 60, 70–78. [98]

Liang, P. P., Liu, T., Ziyin, L., Allen, N. B., Auerbach, R. P., Brent, D., Salakhutdinov, R., and Morency, L.-P. (2020), "Think Locally, Act Globally: Federated Learning with Local and Global Representations," arXiv preprint arXiv:2001.01523. [93]

Lindstrom, M. J., and Bates, D. M. (1988), "Newton–Raphson and EM Algorithms for Linear Mixed-Effects Models for Repeated-Measures Data," *Journal of the American Statistical Association*, 83, 1014–1022. [95]

Majumdar, A., and Gelfand, A. E. (2007), "Multivariate Spatial Modeling for Geostatistical Data Using Convolved Covariance Functions," *Mathematical Geology*, 39, 225–245. [98]

Mak, S., Sung, C.-L., Wang, X., Yeh, S.-T., Chang, Y.-H., Joseph, V. R., Yang, V., and Wu, C. J. (2018), "An Efficient Surrogate Model for Emulation and Physics Extraction of Large Eddy Simulations," *Journal of the American Statistical Association*, 113, 1443–1456. [90]

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017), "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR. [93]

Moreno-Muñoz, P., Artés, A., and Álvarez, M. (2018), "Heterogeneous Multi-Output Gaussian Process Prediction," in *Advances in Neural Information Processing Systems* (Vol. 31), Curran Associates, Inc. [96,101]

Moreno-Muñoz, P., Artes, A., and Álvarez, M. (2021), "Modular Gaussian Processes for Transfer Learning," in *Advances in Neural Information Processing Systems* (Vol. 34), pp. 24730–24740. Curran Associates, Inc. [97]

Nguyen, T. V., Bonilla, E. V., et al. (2014), "Collaborative Multi-Output Gaussian Processes," in *UAI*, pp. 643–652. Citeseer. [96]

Park, J., Han, D.-J., Choi, M., and Moon, J. (2021), "Sageflow: Robust Federated Learning against both Stragglers and Adversaries," in *Advances in Neural Information Processing Systems* (Vol. 34), pp. 840–851, Curran Associates, Inc. [93]

Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N. D., and Karniadakis, G. E. (2017), "Nonlinear Information Fusion Algorithms for Data-Efficient Multi-Fidelity Modelling," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473, 20160751. [101]

Salimans, T., Kingma, D., and Welling, M. (2015), "Markov Chain Monte Carlo and Variational Inference: Bridging the Gap," in *International Conference on Machine Learning* (Vol. 37), pp. 1218–1226. PMLR. [95]

Sattler, F., Wiedemann, S., Müller, K.-R., and Samek, W. (2019), "Robust and Communication-Efficient Federated Learning from Non-iid Data," *IEEE Transactions on Neural Networks and Learning Systems*, 31, 3400–3413. [93]

Sauer, A., Gramacy, R. B., and Higdon, D. (2023), "Active Learning for Deep Gaussian Process Surrogates," *Technometrics*, 65, 4–18. [101]

Shi, N., and Kontar, R. A. (2022), "Personalized Federated Learning via Domain Adaptation with an Application to Distributed 3d Printing," *Technometrics*, 1–22 (just-accepted). [93]

Snelson, E., and Ghahramani, Z. (2005), "Sparse Gaussian Processes Using Pseudo-Inputs," in *Advances in Neural Information Processing Systems* (Vol. 18), MIT Press. [94]

Son, J., Zhou, Q., Zhou, S., Mao, X., and Salman, M. (2013), "Evaluation and Comparison of Mixed Effects Model based Prognosis for Hard Failure," *IEEE Transactions on Reliability*, 62, 379–394. [100]

Sun, Z., Kairouz, P., Suresh, A. T., and McMahan, H. B. (2019), "Can You Really Backdoor Federated Learning?" arXiv preprint arXiv:1911.07963. [93]

Dinh, C. T., Tran, N. H., and Nguyen, T. D. (2020), "Personalized Federated Learning with Moreau Envelopes," in *Advances in Neural Information Processing Systems* (Vol. 33), pp. 21394–21405. Curran Associates, Inc. [93]

Titsias, M. (2009), "Variational Learning of Inducing Variables in Sparse Gaussian Processes," in *Artificial Intelligence and Statistics*, pp. 567–574. PMLR. [95]

Tresp, V. (2000), "A Bayesian Committee Machine," *Neural Computation*, 12, 2719–2741. [97]

Vapnik, V. (1991), "Principles of Risk Minimization for Learning Theory," in *Advances in Neural Information Processing Systems* (Vol. 4), Morgan-Kaufmann. [92]

Ver Hoef, J. M., and Barry, R. P. (1998), "Constructing and Fitting Models for Cokriging and Multivariable Spatial Prediction," *Journal of Statistical Planning and Inference*, 69, 275–294. [90,94]

Wang, K., Mathews, R., Kiddon, C., Eichner, H., Beaufays, F., and Ramage, D. (2019), "Federated Evaluation of On-device Personalization," arXiv preprint arXiv:1910.10252. [93]

Xie, C., Chen, M., Chen, P.-Y., and Li, B. (2021), "CRFL: Certifiably Robust Federated Learning against Backdoor Attacks," in *International Conference on Machine Learning* (Vol. 139), pp. 11372–11382. PMLR. [93]

Yu, H., Guo, K., Karami, M., Chen, X., Zhang, G., and Poupart, P. (2022), "Federated Bayesian Neural Regression: A Scalable Global Federated Gaussian Process," arXiv preprint arXiv:2206.06357. [97]

Yue, X., and Al Kontar, R. (2021), "An Alternative Gaussian Process Objective based on the Rényi Divergence," preprint. [95]

Yue, X., and Kontar, R. A. (2021), "Federated Gaussian Process: Convergence, Automatic Personalization and Multi-Fidelity Modeling," arXiv preprint arXiv:2111.14008. [96,97]

Yue, X., Nouiehed, M., and Al Kontar, R. (2022), "GIFAIR-FL: A Framework for Group and Individual Fairness in Federated Learning," *INFORMS Journal on Data Science*, early access. [93]

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. (2018), "Federated Learning with Non-*i.i.d* Data," arXiv preprint arXiv:1806.00582. [93]

Zhu, H., Xu, J., Liu, S., and Jin, Y. (2021), "Federated Learning on Non-*i.i.d* Data: A Survey," *Neurocomputing*, 465, 371–390. [93]