Bayesian Inference Federated Learning for Heart Rate Prediction

Lei Fang[1], Xiaoli Liu[2], Xiang Su[2,3], Juan Ye[1], Simon Dobson[1], Pan Hui[2,4], and Sasu Tarkoma[2]

[1] University of St Andrews, St Andrews, KY16 9SX, UK lf28@st-andrews.ac.uk
[2] University of Helsinki, Helsinki, 00014, Finland
[3] University of Oulu, Oulu, 90014, Finland
[4] The Hong Kong University of Science and Technology, Hong Kong

Abstract. The advances of sensing and computing technologies pave the way to develop novel applications and services for wearable devices. For example, wearable devices measure heart rate, which accurately reflects the intensity of physical exercise. Therefore, heart rate prediction from wearable devices benefits users with optimization of the training process. Conventionally, Cloud collects user data from wearable devices and conducts inference. However, this paradigm introduces significant privacy concerns. Federated learning is an emerging paradigm that enhances user privacy by remaining the majority of personal data on users' devices. In this paper, we propose a statistically sound, Bayesian inference federated learning for heart rate prediction with autoregression with exogenous variable (ARX) model. The proposed privacy-preserving method achieves accurate and robust heart rate prediction. To validate our method, we conduct extensive experiments with real-world outdoor running exercise data collected from wearable devices.

Keywords: Federated learning · Bayesian inference · Wearable computing · Heart rate prediction

1 Introduction

Cardiovascular diseases (CVD) are the number one cause of death globally. According to the world health organization report, 17.9 million people die from CVD each year, an estimated 31% of all deaths worldwide [1]. Many factors can trigger these diseases, including tobacco use, unhealthy diet, physical inactivity, and harmful use of alcohol. Preventing CVD is becoming an urgent task. It is well-known that exercising has a proven therapeutic effect on the cardiovascular system. Hence, predicting and controlling heart rates during the exercise is important to avoid overstrain and prevent sudden heart rate break.

Wearable devices enable intelligent human-computer interactions. The wearable fitness, sport technologies, and service business are expected to grow exponentially in the near future. Users of wearable devices are expecting the service that can guide their smart exercise coaching, rather than only tracking their

activities. Heart rate based training is a well-known technique to improve the effectiveness of training and prevent over-exercising. Designing an optimal exercise training plan to avoid overstrain is crucial. Ignoring the limits of the physical activities will not only nullify the effect of the exercise but also cause harmful effect on the cardiovascular system. The first step of designing the optimal exercise training plan is predicting heart rate from the exercise, which will be then used for the training control. The designed recommendation and control systems can be adopted in the mobile phone or smart watches. Subjects can use the control system in those smart devices to guide their exercise in order to reach the desired heart rate response and avoid overtraining, which will benefit the users' health. However, most existing research work related to heart rate prediction focuses on indoor exercises. For outdoor physical exercise, it is not possible to automatically regulate the workload intensity due to the dependence on environmental conditions. Hence, it's typical for an outdoor exerciser to continuously check heart rate and increase or decrease the speed accordingly for regulating his or her heart rate.

Machine learning has demonstrated its promising performance in providing the users with recommendations regarding to physical activity and physiological response [2, 3]. Machine learning algorithms typically learn from centralized data in order to train a powerful model. However, pooling data from many users to the Cloud introduces significant privacy concern; for example, leaking sensitive health information of users. Recently, EU General Data Protection Regulation (GDPR) [4] states the need for trust to be built into personal data services and allows users to control their own data, including data their devices generate. Based on GDPR, collecting a massive amount of user data from wearables is not allowed. Federated learning has been regarded as a promising architecture, allowing learning from a large volume of distributed local data without pooling users' private data to the Cloud [5]. Federated learning preserves the users' privacy by training the model in a decentralized manner where multiple local models are synthesized to a global model which is used for future applications.

In this paper, we propose a Bayesian inference based federated learning for heart rate prediction. Bayesian inference provides a statistically sound way to combine local models; and at the same time achieves robust predictions even when data are unevenly distributed among the peripheral nodes, which is common in real world applications. Our work is the first Bayesian inference federated learning approach for heart rate prediction with autoregression with exogenous variable model (ARX) and this framework can be extended to other ARX prediction problems.

Our contributions are threefold:

- We propose two Bayesian federated learning methods, namely Federated Learning based on Sequential Bayesian method (*FD Seq Bayes*) and the Empirical Bayes based Hierarchical Bayesian method (*FD HBayes-EB*), for heart rate prediction without pooling data to the Cloud for privacy preservation. The former model *FD Seq Bayes* is proposed to provide a statistically sound way of integrating local models; whereas the latter model, *FD*

HBayes-EB, provides an alternative but more scalable way from a Bayesian hierarchical model perspective.
- We have conducted extensive evaluation on real-world data from wearable devices. Compared to various state-of-the-art baseline models, our proposed methods have demonstrated their strength in achieving higher prediction accuracy on unseen, new users with lower computation cost.
- Our proposed Bayesian federated learning methods can be easily extended to address other ARX regression problems taking consideration of user privacy preservation and achieving good performance.

The remainder of this paper is organized as follows. Section 2 describes the background and related work. Section 3 presents the proposed Bayesian inference federated learning methods. We present experimentation setup and results in Section 4 and summarize our insights and conclude the paper in Section 5.

2 Related Work

In this section, we review the state-of-the-art techniques in heart rate prediction and federated learning.

2.1 Heart Rate Prediction

Heart rate modelling and prediction have been extensively studied. Existing approaches to model and predict the heart rate response to running exercises can be divided into two categories: (1) Physiological models, which are usually described by deterministic mathematical formulas and used in specific biological systems; and (2) machine learning approaches, which do not encode any prior information but will learn and generalize the response model in the learning process. While approaches in the first category gain its appeal from its analytical closed-form notation, the approaches in the second category are more attractive, because they allow accounting for environmental parameters and other relevant information that is not represented in the analytic equations.

An ordinary differential equation (ODE) model had been proposed by Cheng et al. [6] to describe the dynamical changes of heart rate from resting heart rate by taking consideration of exercise speed and heart rate effects from hormonal system. Levenberg-Marquardt algorithm is used for estimating the optimized parameters. The proposed ODE model is designed for speed control in the treadmill for heart rate regulation. In order to use those models, the subject's resting heart rate need to be known beforehand and special test need to be performed in order to get subject's resting heart rate.

A nonparametric hammerstein model decoupled the linear and nonlinear parts using pseudorandom binary sequences is proposed by Su et.al [7] for heart rate regulation. Support vector regression is adopted to estimate the parameters of the model. Mohammad et al. [8] have used takagi-sugeno fuzzy model for controlling the heart rate in cycling exercises. They build a takagi-sugeno fuzzy

model for each subject based on that subject's own observed data. Subjects did not share their data nor model parameters.

Machine learning methods, such as time series linear regression, support vector regression, feedforward artificial neural network, and long short-term memory (LSTM) [2] have also been used in modelling and predicting the heart rate in exercise. Ni et al. [3] propose an LSTM-based context-aware sequential model to capture the heart rate and the personalized patterns of fitness data. Ludwig et al. [9] summarize most of the recent models related to predicting and controlling heart rate response to exercise.

Current research work related to heart rate modeling and prediction for wearable devices mainly develop general models on the Cloud by sharing subjects' data or developing the personal model with using each subject's own data without sharing other subjects' data. Less attention has been paid on building a general model that can be used for all subjects while keeping data isolated for privacy preservation.

2.2 Federated Learning

Kairouz et al. [10] define federated learning as a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider. Each client's raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective [10]. Federated learning was firstly proposed by Google [5], aiming to keep the training data on the device while collaboratively learning a shared model by coveraging the parameters changes learned from local models. Privacy and communication efficiency are most important concerns in federated learning.

Recently, federated learning has attracted widespread attention and made considerable success in many applications [11]. McMahan et al. [12] have introduced the Federate Averaging (FedAvg) algorithm, which learns the federated global model based on averaging of local learner parameters trained using stochastic gradient descent. Smith et al. [13] treat federated learning as a multi-task learning problem and develop MOCHA method to solve the statistical challenges in federated setting. More significant research work on distributed deep learning can refer to [14, 15]. Chen et al. [16] develop a federated transfer learning framework, named FedHealth, for wearable healthcare. Their proposed approaches combine transfer learning and federated learning using the FedAvg algorithm, which requires to share the same random initialization and is not applicable for combing pre-trained models. Here, we look into a Bayesian model for integrating local models.

Yurochkin et al. [17] propose a Bayesian federated learning framework to aggregate pre-trained neural networks, each being trained locally in parallel with its own specific dataset. The parameters of these local neural networks will be matched to a global model, which is governed by the posterior of a Bayesian nonparametric model. Different from existing work, we focus on learning time-series data with ARX model and we propose two variants of Bayesian methods.

3 Proposed Approach

This section presents the problem statement on federated learning for heart rate prediction and introduces two Bayesian-based techniques: sequential and hierarchical models.

3.1 Problem Definition

The objective of heart rate prediction is to predict the heart rate y_t given the historic readings of the previous heart rates and other useful inputs like speed:

$$y_t = f(y_{1:t-1}, x_{1:t}) + e_t,$$

where e_t is the error term and f can be any parametric function, say a linear function or neural network. The objective of federated learning is to learn such a parametric model f in the server without sending each user's raw data. In particular, given data from n different users stored at each distributed node, and denote the reading from user i as $\mathcal{D}_i$, the learning outcome is a trained global model in the server with datasets $\{\mathcal{D}_i\}_{i=1}^n$; and the global learning should only involve model parameters rather than raw user data. For later prediction, the trained model at the server can then be directly used for predictions of future users with personalization if possible.

3.2 Autoregression with Exogenous Variable Model

As the heart rate data is a time series with serial correlations, a suitable model for such data sets is ARX. An ARX model with p autoregression components and $q+1$ lagged inputs can be formally written as:

$$y_t = \theta_0 + \sum_{i=1}^{p} \theta_i y_{t-i} + \sum_{j=0}^{q} \omega_j z_{t-j} + e_t$$

where $e_t \sim N(0, \sigma^2)$ is white noise with variance σ^2, y_t, z_t are heart rate and speed measurements at time t. By defining $\boldsymbol{\beta}, \boldsymbol{x}$ as the vectors concatenating the model parameters and covariates, the model can be succinctly written as

$$y_t = \boldsymbol{x}^T \boldsymbol{\beta} + e_t,$$

where $\boldsymbol{\beta}^T = [\theta_0, \theta_1, \ldots, \theta_p, \omega_0, \ldots, \omega_q]$ and $\boldsymbol{x}^T = [1, y_{t-1}, \ldots, y_{t-p}, x_t, \ldots, x_{t-q}]$.

3.3 Federated Learning with Sequential Bayesian Inference

Bayesian inference provides a natural solution to the federated learning problem, where the inference is on the posterior distribution of model parameters. By making conditional independent assumption of the data at different nodes given the model parameter, the posterior distribution of the model parameter can be

learnt in a sequential manner. Denoting $\mathcal{D}_i = \{\boldsymbol{X}_i, \boldsymbol{y}_i\}$ as the dataset at node i where $\boldsymbol{y}_i = [y_{i,1}, \ldots, y_{i,n_i}]^T$ and $\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n_i}]^T$, and n_i is the number of time instances for user i; then the posterior distribution is

$$p(\boldsymbol{\beta}, \sigma^2 | \mathcal{D}_1, \mathcal{D}_2, \ldots \mathcal{D}_n) \propto p(\boldsymbol{\beta}, \sigma^2) p(\mathcal{D}_1, \mathcal{D}_2, \ldots \mathcal{D}_n | \boldsymbol{\beta}, \sigma^2) \tag{1}$$

$$= p(\boldsymbol{\beta}, \sigma^2) \prod_{i=1}^{n} p(\mathcal{D}_i | \boldsymbol{\beta}, \sigma^2) \propto p(\boldsymbol{\beta}, \sigma^2 | \mathcal{D}_1, \mathcal{D}_2, \ldots \mathcal{D}_{n-1}) p(\mathcal{D}_n | \boldsymbol{\beta}, \sigma^2), \tag{2}$$

where the second equation has used the conditional independence assumption and the last equation shows that the posterior can be recursively learnt by updating the posterior of the previous $n-1$ sites.

For an ARX model with fixed p, q terms, the model parameters are $\boldsymbol{\beta}$ and σ^2. A conjugate prior for the unknown parameters are Normal-Inverse Gamma distribution, *i.e.*

$$p(\boldsymbol{\beta}, \sigma^2) = \text{NIG}(\boldsymbol{\beta}, \sigma^2; \boldsymbol{m}_0, \boldsymbol{\Lambda}_0, a_0, b_0) \tag{3a}$$

$$= \text{N}(\boldsymbol{\beta}; m_0, \sigma^2 \boldsymbol{\Lambda}_0^{-1}) \text{Inv-Gamma}(\sigma^2; a_0, b_0), \tag{3b}$$

where $\text{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a Gaussian distribution with mean and variance $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$; and Inv-Gamma(a, b) denotes a inverse Gamma distribution with shape and rate parameter a, b respectively.

It can be shown that the posterior distribution can be obtained recursively in a closed form by updating the prior parameters, $\{\boldsymbol{m}_0, \boldsymbol{\Lambda}_0, a_0, b_0\}$, and the inference result is summarized in Theorem 1. According to the update procedure in Equation (5a), instead of averaging all the model parameters learnt at different sites, the Bayesian method essentially provides an alternative weighted average procedure that takes into account of the model uncertainties as well as the parameters themselves. That is, the weights depend on the variance $\boldsymbol{\Lambda}_n^{-1}$, indicating the uncertainty of the parameter.

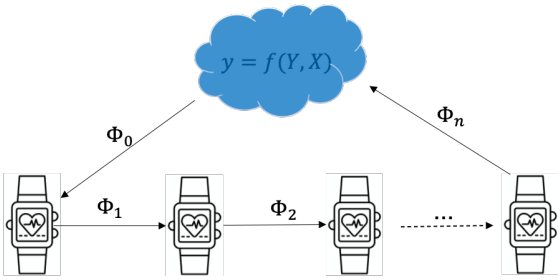

Fig. 1: Federated learning with sequential Bayesian inference

Theorem 1 (Sequential Bayesian inference). *Adopt the NIG prior for $\boldsymbol{\beta}, \sigma^2 | \emptyset$ as defined in Eq. (3) for some pre-determined parameters $\boldsymbol{m}_0, \boldsymbol{\Lambda}_0, a_0, b_0$; for user*

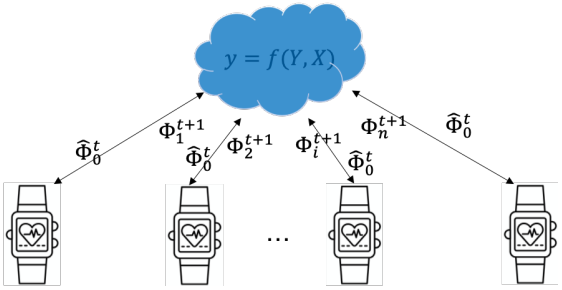

Fig. 2: Federated learning with hierarchical Bayesian inference and empirical Bayes method

datasets $\{\mathcal{D}_i\}_{i=1}^n$*, the posterior distribution can be learnt sequentially, i.e. for* $n > 0$*:*

$$p(\boldsymbol{\beta}, \sigma^2 | \mathcal{D}_1, \ldots, \mathcal{D}_n) = NIG(\boldsymbol{\beta}, \sigma^2; m_n, \boldsymbol{\Lambda}_n, a_n, b_n), \tag{4}$$

where,

$$m_n = (\boldsymbol{\Lambda}_n)^{-1}(\boldsymbol{\Lambda}_{n-1}\boldsymbol{m}_{n-1} + \boldsymbol{X}_n^T \boldsymbol{y}_n) \tag{5a}$$

$$\boldsymbol{\Lambda}_n = \boldsymbol{X}_n^T \boldsymbol{X}_n + \boldsymbol{\Lambda}_{n-1} \tag{5b}$$

$$a_n = a_{n-1} + \frac{N_n}{2} \tag{5c}$$

$$b_n = b_{n-1} + \frac{1}{2}(\boldsymbol{y}_n^T \boldsymbol{y}_n + \boldsymbol{m}_{n-1}^T \boldsymbol{\Lambda}_{n-1} \boldsymbol{m}_{n-1} - \boldsymbol{m}_n^T \boldsymbol{\Lambda}_n \boldsymbol{m}_n) \tag{5d}$$

and N_n *is the number of data points at site* n*.*

To protect the privacy of the user, instead of sending all the raw data $\{\mathcal{D}_i\}$ to the server, we carry out the inference locally in a sequential manner. Each node will learn the posterior sequentially, where the posterior parameters are communicated. To achieve this, a pre-fixed sequential update order needs to be decided at the server and the learning is done essentially by circulating the posterior parameters among the sites. To be more specific, after an update sequential order is initialized, each node i will first receive the model parameters $\boldsymbol{\Phi}_{i-1} = \{\boldsymbol{m}_{i-1}, \boldsymbol{\Lambda}_{i-1}, a_{i-1}, b_{i-1}\}$ from the previous user, or the server if it is the first update iteration, $i = 1$. Then the parameters will be updated according to Theorem 1. The updated parameters $\boldsymbol{\Phi}_i = \{\boldsymbol{m}_i, \boldsymbol{\Lambda}_i, a_i, b_i\}$ will be relayed to the next node $i+1$ until all sites update their parameters. The server will receive and keep the learnt posterior parameter $\boldsymbol{\Phi}_n = \{\boldsymbol{m}_n, \boldsymbol{\Lambda}_n, a_n, b_n\}$ for later prediction. The learning procedure is described in Fig. 1.

3.4 Federated Learning with Hierarchical Bayesian Inference

The sequential processing algorithm clearly does not scale well when the number of users/nodes increases. A distributed inference algorithm that allows parallel

processing therefore is more appealing. To achieve this, hierarchical Bayesian model is proposed, where hyper-priors over the prior parameters are introduced and the model parameters at each individual site become conditionally independent. By using a hierarchical model, we also achieve a principled way of learning hyperparameters, $\{\boldsymbol{m}_0, \boldsymbol{\Lambda}_0, a_0, b_0\}$.

Formally, a hierarchical Bayesian linear regression model can be formulated as follows

$$\begin{aligned} \boldsymbol{y}_i|\boldsymbol{\beta}_i, \sigma_i^2, \boldsymbol{X}_i &\sim \mathrm{N}(\boldsymbol{X}_i\boldsymbol{\beta}_i, \sigma_i^2\boldsymbol{I}) \\ \boldsymbol{\beta}_i, \sigma_i^2|\boldsymbol{m}_0, \boldsymbol{\Lambda}_0, a_0, b_0 &\sim \mathrm{NIG}(\boldsymbol{m}_0, \boldsymbol{\Lambda}_0, a_0, b_0) \\ \boldsymbol{m}_0, \boldsymbol{\Lambda}_0, a_0, b_0|\boldsymbol{\Psi} &\sim P(.), \end{aligned}$$

where each user/node has its own model parameter $\{\boldsymbol{\beta}_i, \sigma_i^2\}$. A common Normal-InvGamma prior is imposed on the model parameters and the model parameters become conditionally independent or exchangeable given the hyperparameters $\boldsymbol{\Phi}_0 = \{\boldsymbol{m}_0, \boldsymbol{\Lambda}_0, a_0, b_0\}$. A further hierarchical hyper-prior P of appropriate form is imposed on the hyperparameters $\boldsymbol{\Phi}_0$. For example, Gaussian is for $\boldsymbol{m}_0$, Inverse-Wishart is for $\boldsymbol{\Lambda}_0$, and Gamma is for a_0 and b_0. Usually vague uninformative hyper-priors are used for the second tier distributions [18].

Empirical Bayes The inference for the hierarchical model cannot be solved in closed form any more. Usually computationally expensive inference procedures like Markov Chain Monte Carlo (MCMC) has to be used. An alternative is Empirical Bayes (EB) method where hyperparameters $\boldsymbol{\Phi}_0 = \{\boldsymbol{m}_0, \boldsymbol{\Lambda}_0, a_0, b_0\}$ are not sampled but directly maximised against the model evidence

$$\hat{\boldsymbol{\Phi}}_0 = \underset{\boldsymbol{\Phi}_0}{\operatorname{argmax}}\, P(\mathcal{D}_1, \mathcal{D}_2 \ldots \mathcal{D}_n|\boldsymbol{m}_0, \boldsymbol{\Lambda}_0, a_0, b_0).$$

By treating the model parameters $\{\boldsymbol{\beta}_i, \sigma_i^2\}_{i=1}^n$ as missing data, an EM algorithm can be derived to find the optimal hyperparameters. The detailed derivation and the EM algorithm is listed in the appendix.

The EB-based federated learning becomes an iterative procedure to accommodate the learning of the hyperparameters $\boldsymbol{\Phi}_0$. The learning procedure iterates between the following two steps:

1. Update the hyperparameter at the server given $P(\{\boldsymbol{\beta}_i, \sigma_i^2\}_{i=1}^n|\{\mathcal{D}_i\}_{i=1}^n, \hat{\boldsymbol{\Phi}}_0^{t-1})$ by an EM procedure listed Eq. (9):

$$\hat{\boldsymbol{\Phi}}_0^t = \underset{\boldsymbol{\Phi}_0}{\operatorname{argmax}}\, P(\mathcal{D}_1, \mathcal{D}_2 \ldots \mathcal{D}_n|\boldsymbol{m}_0, \boldsymbol{\Lambda}_0, a_0, b_0).$$

2. At each site i, update the local posterior given $\hat{\boldsymbol{\Phi}}_0^t = \{\hat{\boldsymbol{m}}_0, \hat{\boldsymbol{\Lambda}}_0, \hat{a}_0, \hat{b}_0\}$ in parallel:

$$P(\boldsymbol{\beta}_i, \sigma_i^2|\hat{\boldsymbol{\Phi}}_0^t, \mathcal{D}_i) = \mathrm{NIG}(\boldsymbol{m}_i, \boldsymbol{\Lambda}_i, a_i, b_i), \text{ where} \tag{6a}$$

$$
\begin{aligned}
\boldsymbol{m}_i &= (\boldsymbol{\Lambda}_i)^{-1}(\hat{\boldsymbol{\Lambda}}_0\hat{\boldsymbol{m}}_0 + \boldsymbol{X}_n^T\boldsymbol{y}_n) \\
\boldsymbol{\Lambda}_i &= \boldsymbol{X}_i^T\boldsymbol{X}_i + \hat{\boldsymbol{\Lambda}}_0 \\
a_i &= \hat{a}_0 + \frac{N_i}{2} \\
b_i &= \hat{b}_0 + \frac{1}{2}(\boldsymbol{y}_i^T\boldsymbol{y}_i + \hat{\boldsymbol{m}}_0^T\hat{\boldsymbol{\Lambda}}_0\hat{\boldsymbol{m}}_0 - \boldsymbol{m}_i^T\boldsymbol{\Lambda}_i\boldsymbol{m}_i)
\end{aligned} \tag{6b}
$$

To be more specific, at each iteration $t \geq 1$, the server will propagate the current hyperparameter $\hat{\boldsymbol{\Phi}}_0^{t-1}$ to the clients (some initial non-informative piror's parameters are used for the first iteration), each node then updates their posterior distributions of the local parameters according to a variant of Theorem 1 and sends back the learnt posterior parameters $\boldsymbol{\Phi}_i^t = \{\boldsymbol{m}_i, \boldsymbol{\Lambda}_i, a_i, b_i\}$ to the server. The server will then optimize the hyperparameter $\hat{\boldsymbol{\Phi}}_0^t$ based on the received posterior distributions by the EM algorithm. Fig. 2 summarizes the learning procedure at iteration t. Note that the local learning in Eq (6) is still in closed form hence computationally cheap. Thanks to this conjugacy, we find that only two to three iterations are usually good enough for the EB method to work in practice.

4 Evaluation and Results

This section illustrates our evaluation methodology, including the dataset and comparison techniques, and then present the results.

4.1 Dataset

We analyze the performance of proposed Bayesian inference federated learning with collecting real-world outdoor running exercise data from 10 subjects wearing Polar smart watches. Exercise time, running speed, and heart rate are recorded in each exercise. The physical characteristics of subjects are listed in Table 1. The duration of one exercise ranges from 30 minutes to 90 minutes and heart rate ranges from 60bpm to 200bpm. Outliers are removed based on the interquartile range criteria and missing values are imputed with linear interpolation. The ten subjects are regarded as isolated to each other and cannot share their data due to the privacy concern during the federated learning process.

4.2 Evaluation Methods

We evaluate the two proposed Bayesian Federated learning, *FD Seq Bayes* and *FD HBayes-EB*, with two baseline solutions.

- FedAvg: a simple average based federated learning method where the local regression models are trained by the least squared error method. A simple average of the learnt parameters is used for future testing and prediction.

Table 1: Physical characteristics of the subjects.

	Age(yr)	Height(cm)	Weight(kg)	BMI(kg/m2)
Mean	30.4	175.2	70.8	23.05
Standard Deviation	2.5	7.5	9.2	1.6
Range	(27, 34)	(162, 187)	(55, 87)	(19.9, 24.8)

– HBayes-MCMC: It refers to hierarchical Bayesian model inferred by Markov Chain Monte Carlo (MCMC) method [18]. Note that this method does not belong to federated learning realm as the training data from all the users is aggregated and stored in the server.

4.3 Experiment Procedure

To evaluate the effects of the proposed methods thoroughly, we firstly randomly select nine out of ten users as the existing users, leaving one user's data for testing as new users. For the nine chosen users, a random subset of each user's K_i exercises data is selected for model training: the selected exercises data is further split into training and testing. The following three types of errors are compared, including training error, testing error, and new exercise error (on the left-out user's and the unselected exercises' data). We assume that the training and testing data are drawn from the same population, and thus the training and testing errors are used to assess the learning capability of the model. The new exercise error represents the model performance on new users and new exercises from selected users (which might have different distributions from the training and testing data), indicating the generalization of the model. Squared errors are used for evaluating the performance of the models. The errors are further decomposed as by time-instance error and by-user error. The definitions of these two errors are as follows.

$$\text{error}_{\text{time}} = \frac{\sum_{i=1}^{n}\sum_{t=1}^{n_i}(y_{i,t} - \hat{y}_{,it})^2}{\sum_{i=1}^{n} n_i},$$

$$\text{error}_{\text{user}} = \frac{\sum_{i=1}^{n}(\sum_{t=1}^{n_i}(y_{i,t} - \hat{y}_{i,t})^2/n_i)}{n},$$

where n is the total number of users (10 users in our case) and n_i is the number of data records of user i's data.

4.4 Results

Table 2 and Fig. 3 report the experiment results of $K_i = 10$ for 100 repeated experiments. Table 2 reports the means of squared errors and standard deviations of means. As we can see that the proposed two federated learning methods, i.e.

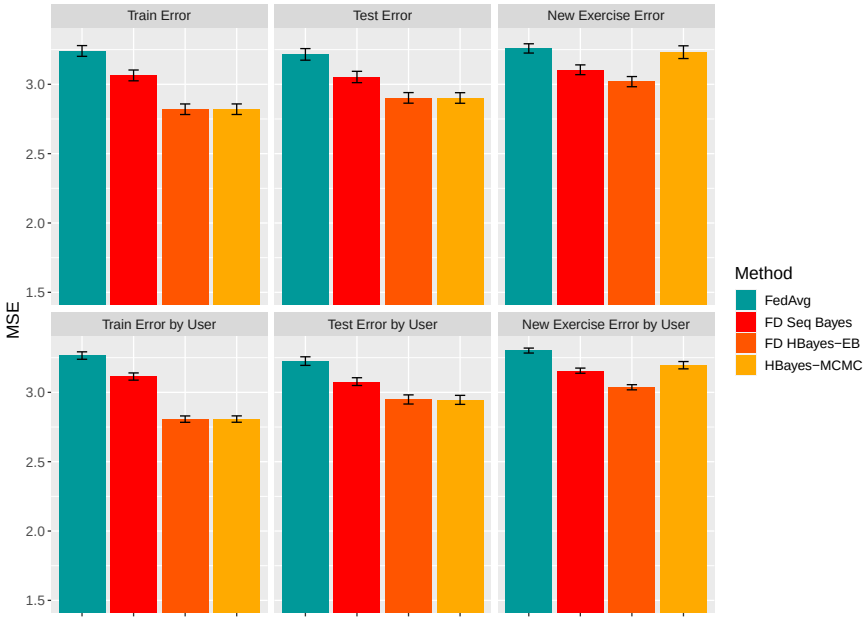

Fig. 3: Experiment results on the four methods; where $K_i = 10$ exercises data are used for training; where the error bars are the standard error

FD Seq Bayes and FD HBayes-EB outperform the simple FedAvg by significant margins in all six types of errors; *e.g.*, FD Seq Bayes and FD HBayes-EB reduce 0.16 and 0.46 on the mean of train error and 0.15 and 0.28 on the mean of test error from FedAvg.

The hierarchical Bayesian method achieves better result compared to the sequential method. The empirical Bayes method also achieves very similar results compared to the more computation intensive MCMC-based method in both training and testing errors and outperforms its counterpart in the new exercise error. Our results show that Bayesian based federated learning methods provide a more sound model synthesis (smaller testing error) and also new user personalization performance (new exercise error).

Effects of Varied Training Datasets To further demonstrate the effects of the Bayesian inference federated learning method, we deliberately make the training data imbalanced to better simulate the real world scenario. To be more specific, a random selection ratio $(0.01\%, 25\%, 50\%, 75\%, 100\%)$ is applied to each user's training data to make the training data imbalanced among the users. We assume different amounts of data from users might have a negative impact on federated learning; for example, the prediction might be biased towards the users whose data takes the majority. The results of 100 random experiments are listed in Fig. 4. It is obvious that all Bayesian based methods, both feder-

Table 2: Experiment results of by user error on the four methods; where $K_i = 10$ exercises data are used for training. The mean and the standard deviations (in brackets) are reported.

	Train Error by User	Test Error by User	New Ex Error by User
FedAvg	3.27 (0.27)	3.23 (0.31)	3.3 (0.18)
FD Seq Bayes	3.11 (0.26)	3.08 (0.28)	3.16 (0.19)
FD HBayes-EB	2.81 (0.23)	2.95 (0.33)	3.04 (0.19)
HBayes-MCMC	2.81 (0.23)	2.95 (0.33)	3.2 (0.26)

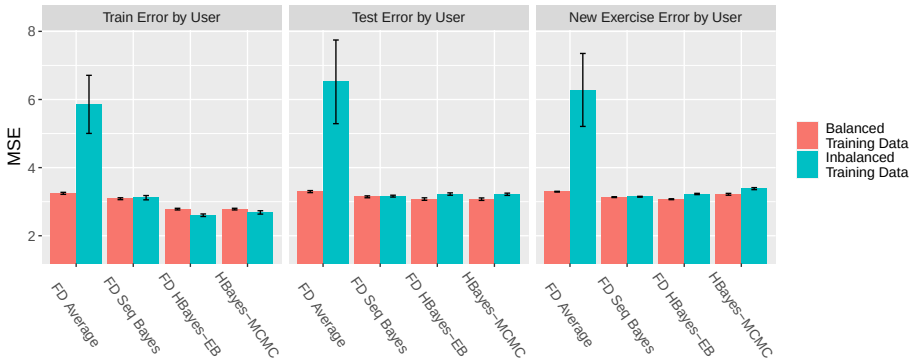

Fig. 4: Experiments on imbalanced training data scenario

ated and traditional learning, outperform the likelihood based average method. When imbalanced training data is used, the average methods fail in all three error categories, and their large standard error also signifies the instability of the methods. The Bayesian methods however are all more robust, i.e. the performance deteriorates to a much less degree. We can also observe that the sequential Bayesian method achieves slightly better results than the hierarchical model, as the sequential method essentially pools the data together by integrating all local parameters with weights but at the price of scalability.

5 Conclusion

When users perform physical exercises, one important goal is to optimize the training process. Heart rate has been used as a most important indicator for monitoring the training strain. Therefore, predicting heart rate during physical exercise is crucial for tracking physiological responses and improving the effect of the exercise. The majority of the existing research focuses on pooling together a large amount of users' data for building a robust model, which often has incurred much privacy concern. To tackle this issue, we leverage a statistically sound model – Bayesian inference and propose two Bayesian-based federated learning

methods, i.e. *FD Seq Bayes* and *FD HBayes-EB*. They enable collaborative model training under the orchestration of a central server, while not accessing to any user's local data. Through extensive evaluation on real-world dataset, we have demonstrated the advantages of our methods in accurate prediction and low computation cost. In the future, we will extend our evaluation to other ARX regression problems to assess the generalization of our methods.

6 Acknowledge

This work has been partially supported by the UK EPSRC under grant number EP/N007565/1, "Science of Sensor Systems Software", and by Academy of Finland projects, grant number 325774, 3196669, 319670, 326305, and 325570.

A EM algorithm for hyperparameter estimation for hierarchical Bayesian regression model

E step: The complete data log likelihood is

$$\begin{aligned} L(\boldsymbol{\Phi}_0) &= \log P(\{\boldsymbol{\beta}_i, \sigma_i^2\}_1^n, \{\mathcal{D}_i\}_i^n | \boldsymbol{\Phi}_0) \\ &= \log(P(\{\mathcal{D}_i\}_i^n | \{\boldsymbol{\beta}_i, \sigma_i^2\}_1^n, \boldsymbol{\Phi}_0) P(\{\boldsymbol{\beta}_i, \sigma_i^2\}_1^n | \boldsymbol{\Phi}_0)) \\ &= \log \left(\prod_{i=1}^{n} \mathrm{N}(\boldsymbol{y}_i; \boldsymbol{X}_i \boldsymbol{\beta}_i, \sigma_i^2 \boldsymbol{I}) \mathrm{NIG}\left(\{\boldsymbol{\beta}_i, \sigma_i^2\}; \boldsymbol{\Phi}_0\right) \right) \\ &= \sum_{i=1}^{n} \log\left(\mathrm{NIG}\left(\{\boldsymbol{\beta}_i, \sigma_i^2\}; \boldsymbol{\Phi}_0\right)\right) + C, \end{aligned}$$

where C contains all the terms that are independent of $\boldsymbol{\Phi}_0$. The conditional expected complete data likelihood is:

$$\begin{aligned} Q(\boldsymbol{\Phi}_0 | \boldsymbol{\Phi}_0^{t-1}) &= E_{\{\boldsymbol{\beta}_i, \sigma_i^2\}_1^n | \boldsymbol{\Phi}_0^{t-1}, \{\mathcal{D}_i\}_i^n}[L(\boldsymbol{\Phi}_0)] \\ &\approx \frac{1}{nL} \sum_{m=1}^{L} \sum_{i=1}^{n} \log(\mathrm{NIG}(\{\boldsymbol{\beta}_i, \sigma_i^2\}^{(m)}; \boldsymbol{\Phi}_0)) \end{aligned}$$

where $\{\boldsymbol{\beta}_i, \sigma_i^2\}^{(m)}$ denotes the $m-$th i.i.d. sample from $P(\boldsymbol{\beta}_i, \sigma_i^2 | \mathcal{D}_i, \boldsymbol{\Phi}_0^{t-1})$, which are NIG distributed. Sampling from a NIG distribution is straightforward by a standard two step procedure by firstly sampling σ^2 from Inv-Gamma(a_i, b_i) then sampling from $\boldsymbol{\beta}$ from $\mathrm{N}(\boldsymbol{m}_i, \sigma^2 \boldsymbol{\Lambda}_i^{-1})$. Essentially, we are approximating the conditional expectation with a Monte Carlo estimator with L samples from the posterior $P(\{\boldsymbol{\beta}_i, \sigma_i^2\}_1^n | \{\mathcal{D}_i\}_1^n, \boldsymbol{\Phi}_0^{t-1})$. The EM algorithm degenerates to a Monte Carlo Expectation Maximization (MCEM) [19].

M step: the objective here is to maximize the conditional expectation, namely

$$\hat{\boldsymbol{\Phi}}_0 = \underset{\boldsymbol{\Phi}_0}{\operatorname{argmax}}\, Q(\boldsymbol{\Phi}_0|\boldsymbol{\Phi}_0^{t-1}) \tag{7}$$

$$= \underset{\boldsymbol{\Phi}_0}{\operatorname{argmax}} \frac{1}{nL} \sum_{m=1}^{L}\sum_{i=1}^{n} \log(\mathrm{N}(\boldsymbol{\beta}_i^{(m)}; \boldsymbol{m}_0, \sigma_i^{2(m)}\boldsymbol{\Lambda}_0^{-1})) + \log\left(\mathrm{G}\left(\sigma_i^{-2(m)}; a_0, b_0\right)\right) \tag{8}$$

where we have used the property that if $x \sim$ Inv-Gamma(a, b), then $1/x$ is Gamma distributed with shape and rate parameters a, b, denoted as $\mathrm{G}(a, b)$. It is easy to see that the optimal $\hat{a}_0, \hat{b}_0$ w.r.t Q are just the maximum likelihood estimator of a Gamma distribution with dataset $\{\sigma_i^{2(m)}\}_{i,m=1}^{n,L}$ (the second term of Equation (8)). An iterative generalized Newton's method can be used to find the ML estimator of Gamma as follows [20].

$$\frac{1}{a_0} = \frac{1}{a_0} + \frac{\overline{\log\sigma^{-2}} - \log(\overline{\sigma^{-2}}) + \log a_0 - \Psi(a_0)}{a_0^2(1/a_0 - \Psi'(a_0))} \tag{9a}$$

$$b_0 = \frac{\overline{\sigma^{-2}}}{a_0}, \tag{9b}$$

where

$$\overline{\sigma^{-2}} = \frac{\sum_{m=1}^{L}\sum_{i=1}^{n} 1/\sigma_i^{2(m)}}{nL},\ \overline{\log\sigma^{-2}} = \frac{\sum_{m=1}^{L}\sum_{i=1}^{n} \log(1/\sigma_i^{2(m)})}{nL}.$$

Take the derivative of the Gaussian term in Eq. (8) w.r.t $\boldsymbol{m}_0, \boldsymbol{\Lambda}_0$ and set them to zero, we can find the estimators for $\boldsymbol{m}_0, \boldsymbol{\Lambda}_0$:

$$\boldsymbol{m}_0 = \frac{\sum_{m=1}^{L}\sum_{i=1}^{n} \frac{1}{\sigma_i^{2(m)}}\boldsymbol{\beta}_i^{(m)}}{\sum_{m=1}^{L}\sum_{i=1}^{n} \frac{1}{\sigma_i^{2(m)}}} \tag{9c}$$

$$\boldsymbol{\Lambda}_0^{-1} = \frac{1}{nL}\sum_{m=1}^{L}\sum_{i=1}^{n} \frac{1}{\sigma_i^{2(m)}}(\boldsymbol{\beta}_i^{(m)} - \boldsymbol{m}_0)(\boldsymbol{\beta}_i^{(m)} - \boldsymbol{m}_0)^T \tag{9d}$$

References

1. Cardiovascular diseases https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1. Last accessed 22 Aug 2020.
2. Hilmkil, A., Ivarsson, O., Johansson, M., Kuylenstierna, D., Erp, T. V.: Towards Machine Learning on data from Professional Cyclists. In: 12th World Congress on Performance Analysis of Sports, Opatija, Croatia (2018).
3. Ni, J. M., Muhlstein, L., McAuley, J.: Modeling Heart Rate and Activity Data for Personalized Fitness. In: WWW'19, May 13–17, 2019, San Francisco, CA, USA.
4. EU. Regulation (EU) 2016/679 of the European Parliamentand of the Council of 27 April 2016 on the protection of natural persons with regard to the

processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union, L119:1–88,may 2016. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC
5. Konečný J., McMahan H. B., Ramage D., Richtárik P.: Federated Optimization: Distributed Machine Learning for On-Device Intelligence. arXiv:1610.02527 (2016).
6. Cheng, T. M., Savkin, A. V., Celler, B. G., Su, S. W., Wang, L.: Nonlinear modeling and control of human heart rate response during exercise with various work load intensities. IEEE Trans. Biomed. Eng. **55**(11), 2499–2508 (2008).
7. Su, S. W., Wang, L., Celler, B. G., Savkin, A. V., Guo, Y.: Identification and control for heart rate regulation during treadmill exercise. IEEE Trans. Biomed. Eng. **54**(7), 1238–1246 (2007b).
8. Mohammad, S., Guerra, T. M., Grobois,J. M., Hecquet, B.: Heart rate control during cycling exercise using Takagi-Sugeno models. In: 8th IFAC World Congress, pp. 12783–12788, Milano, Italy (2011).
9. Ludwig, M., Hoffmann, K., Endler, S., Asteroth, A., Wiemeyer, j.: Measurement, Prediction, and Control of Individual Heart Rate Responses to Exercise-Basics and Options for Wearable Devices. Front. Physiol. (2018). https://doi.org/10.3389/fphys.2018.00778
10. Kairouz, P., et al.: Advances and Open Problems in Federated Learning. arXiv:1912.04977 (2019).
11. Xu D. L., Li T., Li Y., Su X., Tarkoma S., Jiang T., Crowcroft J., Hui P.: Edge Intelligence: Architectures, Challenges, and Applications. arXiv:2003.12172 (2020).
12. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B. A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, pp. 1273–1282, 2017.
13. Smith, V., Chiang, C. K., Sanjabi, M., Talwalkar, A. S.: Federated multi-task learning. In: NIPS 2017, pp. 4424–4434, Long Beach, CA, USA (2017).
14. Lian, X., Zhang, C., Zhang, H., Hsieh, C. J., Zhang, W., Liu, J.: Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In: Advances in Neural Information Processing Systems 30, pp. 5330–5340 (2017).
15. Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M.,Mao, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., et al.: Large scale distributed deep networks. In: Advances in neural information processing systems 25, pp. 1223–1231 (2012).
16. Chen, Y., Qin, X., Wang, J., Yu, C., Gao, W.: FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare. IEEE Intelligent Systems (2109).
17. Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N. & Khazaeni, Y.. (2019). Bayesian Nonparametric Federated Learning of Neural Networks. In: PMLR 97:7252-7261.
18. Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB.: Bayesian data analysis. CRC press; (2013).
19. Wei G. C., Tanner, M. A.: A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. Journal of the American statistical Association, 85(411), 699-704 (1990).
20. Minka, T. P.: Estimating a Gamma distribution. Microsoft Research, Cambridge, UK, Tech. Rep (2002).