2025 3rd International Conference on Federated Learning Technologies and Applications (FLTA) | 979-8-3315-5670-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/FLTA67013.2025.11336663

Applying Federated Learning to Block-Term Tensor Regression for Decentralised Data Analysis of Biomedical Data

1st Axel Faes
Biomedical Data Sciences, DSI & BIOMED
UHasselt, Belgium
0000-0002-1637-255X

2nd Ashkan Pirmani
Biomedical Data Sciences, DSI & BIOMED
UHasselt, Belgium
STADIUS, ESAT KU Leuven, Belgium
0000-0003-4549-1002

3th Yves Moreau
STADIUS, ESAT KU Leuven, Belgium
0000-0002-4647-6560

4th Liesbet M. Peeters
Biomedical Data Sciences, DSI & BIOMED
UHasselt, Belgium
0000-0002-6066-3899

***Abstract*—Block-Term Tensor Regression (BTTR) is a powerful method for modeling complex, high-dimensional data through multilinear relationships, making it well-suited for healthcare and neuroscience. However, its reliance on centralized datasets raises privacy concerns and limits collaboration. To address this, we propose Federated Block-Term Tensor Regression (FBTTR), an extension of BTTR for federated learning that enables decentralized model building while preserving privacy and regulatory compliance.**

We evaluate FBTTR in two case studies: finger movement decoding from Electrocorticography (ECoG) signals and disease risk prediction. On the BCI Competition IV dataset, FBTTR outperforms non-multilinear models and achieves higher accuracy than centralized BTTR (e.g., subject 3 thumb decoding: 0.76 ± 0.05 **vs.** 0.71 ± 0.05**). On real-world clinical data, FBTTR surpasses both standard federated approaches and centralized BTTR (e.g., Fed-Heart-Disease Dataset AUC-ROC:** 0.872 ± 0.02 **vs.** 0.812 ± 0.003**; accuracy:** 0.772 ± 0.02 **vs.** 0.753 ± 0.007**).**

These results show that FBTTR is scalable, computationally efficient, and achieves predictive performance comparable to centralized models. Released as open-source software, it provides a practical and transparent framework for advancing federated analytics in healthcare and brain-computer interface applications.

***Index Terms*—federated learning, BTTR, tensor, regression**

I. Introduction

Within regression analysis, multiway approaches are being increasingly applied. For example, they have been used to model arm trajectories from Electrocorticography (ECoG) signals recorded in monkeys, as discussed in [1], [2]. These methods have also been explored in the context of exoskeleton-based arm trajectories, arm and wrist rotations, and ECoG signals in tetraplegic patients [3]. In addition, they have been applied to the study of graded stimulation of the spinal cord regions involved in walking in tetraplegic patients [4].

In the latter two scenarios, multiway partial least squares regression (NPLS) or its variations are used within decoders. However, this approach has not achieved the accuracy required for decoding fine finger movements [3]. This shortcoming may be linked to factors such as limited fitness capacity, high computational complexity, and slow convergence of NPLS, especially when applied to higher-order data [1].

On a more promising note, Zhao et al. [1] introduced a generalized framework known as Higher-Order Partial Least Squares (HOPLS). This framework is based on a $(1, L_2, ..., L_N)$-rank Block Term Decomposition (BTD), in which all blocks share the same multilinear rank (MTR, Multilinear Tensor Rank). HOPLS strikes a balance between model fit and complexity, improving predictability and demonstrating superior performance compared to conventional Partial Least Squares (PLS).

A. Evolution of BTTR

Camarrone et al. introduced Block-Term Tensor Regression (BTTR) [5], [6], which relies on Tucker decomposition combined with a deflation approach to extract successive blocks of decreasing relevance for regression. Each block may have distinct MTR ranks, with parameters optimized block-wise through Automatic Component Extraction (ACE). BTTR achieves results comparable to Higher Order Partial Least Squares (HOPLS), yet with faster training, although it is limited to scalar output prediction. This limitation complicates tasks like decoding complex finger motions, where one option is fitting separate BTTR models for each finger, at the cost of losing shared information. Overlapping and sparse ECoG signals during multi-finger flexions further challenge decoding accuracy [7]. To address this, Faes et al. proposed eBTTR (extended BTTR) [8], which models coordinated finger movements and unintended co-activations, even though ECoG signals may not directly encode them. However, while recursive decomposition in eBTTR improves some single-finger predictions, it can reduce accuracy for others compared to standard BTTR. Both versions of BTTR are fully centralised

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 29,2026 at 07:33:18 UTC from IEEE Xplore. Restrictions apply.

algorithms that require the full dataset to be available at a single location.

B. Federated Learning within Healthcare

Federated learning (FL) is a newer paradigm tackling privacy concerns by decentralizing model training [9]. McMahan et al. introduced Federated Averaging (FedAvg) [10], aggregating local updates to build a global model while minimizing data transfer. Bonawitz et al. enhanced FL privacy through secure aggregation [11]. Privacy remains a core motivation for FL. Geyer et al. developed differentially private FL [12], complementing earlier work on privacy-preserving deep learning by Abadi et al. [13]. Truex et al. proposed combining secure multiparty computation with differential privacy [14]. Scaling FL for practical use remains crucial. Surveys by Kairouz et al. [15] and Yang et al. [16] discuss solutions for efficiency and large-scale deployment, including hierarchical FL frameworks.

Applications of FL span diverse fields. Hard et al. [17] improved mobile keyboard prediction with FL, while Sheller et al. applied FL to medical imaging [18], aiding collaborative research without sharing sensitive data. Despite advances, challenges persist around data heterogeneity, communication bottlenecks, and robustness to attacks [19]. FL enables collaborative model training while preserving data privacy, either by avoiding raw data exchange or via encryption [20]. Vertical FL (VFL) applies when parties share the same samples but hold different features. Cross-organization scenarios fit VFL, since companies often link data via product or batch IDs, yet differ in local features due to distinct processes. Progress on using VFL for process modeling remains limited.

The prevalence of heart disease, one of the leading causes of mortality worldwide, necessitates accurate predictive models to aid in early diagnosis and treatment. Traditional machine learning models often require centralized data, which poses significant privacy risks in healthcare. Tensor regression models, such as the Block-Term Tensor Regression (BTTR), have shown promise in handling multi-dimensional data prevalent in medical records. However, the centralized nature of these models limits their scalability and applicability in real-world healthcare settings. Federated learning offers a solution to these challenges by enabling collaborative model training across multiple institutions without sharing sensitive data. We introduce the FBTTR framework and apply it to various types of biomedical data such as heart disease prediction, leveraging the advantages of tensor regression in a federated context to improve predictive accuracy while ensuring data privacy.

In this paper, we introduce Federated Block-Term Tensor Regression (FBTTR), an extension of BTTR designed for federated learning environments. Federated learning allows models to be trained across multiple decentralized institutions without sharing patient data, thus enhancing privacy. Our objective is to leverage the advantages of BTTR in a federated context to improve heart disease prediction while ensuring data privacy. Empirical evaluations demonstrate the robustness of FBTTR, showing that it achieves high predictive performance comparable to traditional centralized models. Moreover, the algorithm exhibits efficient computational performance, making it suitable for deployment in real-world healthcare environments where timely and accurate predictions are crucial. The results highlight FBTTR's potential in advancing predictive analytics in healthcare by providing a scalable, privacy-preserving solution for multiple domains including heart disease prediction and brain-computer interfaces.

TABLE I
MATHEMATICAL NOTATION

Notation	Description
$\underline{\mathbf{T}}, \mathbf{M}, \mathbf{v}, S$	tensor, matrix, vector, scalar (respectively)
$\mathbf{M}^T$	transpose of matrix
$\times_n$	mode-n product between tensor and matrix
$\otimes$	Kronecker product
$\circ$	outer product
$\|\cdot\|_F$	Frobenius norm
$\mathbf{T}_{(n)}$	mode-n unfolding of tensor $\underline{\mathbf{T}}$
$\underline{\mathbf{C}}^{(T)}$	core tensor associated to tensor $\underline{\mathbf{T}}$
$\mathbf{M}^{(n)}$	mode-n factor matrix
$\mathbf{M}_{ind}$	(sub-)matrix including the column(s) indicated in ind
$\mathbf{M}_{\setminus ind}$	(sub-)matrix excluding the column(s) indicated in ind
$[\![\underline{\mathbf{C}}; \mathbf{M}^{(1)}, ..., \mathbf{M}^{(N)}]\!]$	full multilinear product $\underline{\mathbf{C}} \times_1 \mathbf{M}^{(1)} \times_2 \cdots \times_N \mathbf{M}^{(N)}$
$\langle \underline{\mathbf{T}}, \underline{\mathbf{E}} \rangle_{\{n,n\}}$	mode-n cross-covariance tensor

II. Related Work

A. Block-Term Tensor Regression

Block-Term Tensor Regression (BTTR) decomposes the input tensor into block components, facilitating efficient regression on high-dimensional data. It effectively captures the underlying structure of the data, making it suitable for applications involving multi-modal datasets [6]. An overview of the mathematical notation used can be found in Table I.

B. Federated Learning Definition

Federated learning (FL) is a paradigm where multiple parties train models collaboratively without sharing or centralizing their data. Participants update a global model by performing local optimization and sending updates, although more local training can reduce guarantees of minimizing the overall objective. Aggregation depends on network topology, as nodes may be split into subnetworks due to geographical or legal limits. FL can be classified as horizontal, when datasets share the same features but different samples, like in Google Keyboard, or vertical, when datasets have different features for the same individuals. It can also be synchronous, aggregating updates from clients together, or asynchronous, updating the global model as soon as client updates arrive. Finally, FL may follow a hub-and-spoke model with a central server or a peer-to-peer approach where updates are shared directly between

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 29,2026 at 07:33:18 UTC from IEEE Xplore. Restrictions apply.

participants, often sharing only part of model parameters for efficiency or privacy.

C. Non-multilinear approaches

To provide a comprehensive comparison of the performance of the Federated Block Term Tensor Regression (FBTTR) method with other established and more recent techniques, we will employ the BCI competition IV dataset as one of our case studies (details provided later). Specifically, we will benchmark FBTTR against the competition-winning method, which utilized a linear regression model based on amplitude modulation (AM) [21]. Additionally, we will compare it with more contemporary approaches that have leveraged Random Forests (RF), Convolutional Neural Networks (CNN), and Long Short-Term Memory Networks (LSTM), as proposed and evaluated in [22]. Notably, all of these models fall under the category of non-multilinear approaches.

III. Methodology

A. Federated Block-Term Tensor Regression (FBTTR)

Federated Block-Term Tensor Regression (FBTTR) extends BTTR by integrating it into a federated learning framework. The architecture consists of multiple local models trained as seperate clients. These models are periodically synchronized by aggregating their parameters on a central server. The steps involved in FBTTR are:

- **Data Partitioning and Federation** - The dataset is partitioned across multiple institutions, each retaining its data locally.
- **Local Model Training** - Each client trains a local BTTR model on its data.
- **Model Aggregation** - The local models' parameters are securely aggregated to update the global model.

The main aggregating node, the server, is responsible for coordinating the training process. The server initializes the global model and sends it to the clients. The clients train their local models and send the updates back to the server. The server aggregates the updates and updates the global model. This process is repeated for a specified number of federated rounds. The server then returns the final global model. The algorithm for FBTTR is shown in Algorithm 2. Hereby, $\mathbf{E}, \mathbf{F}$ refer to the input $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}, \mathbf{Y} \in \mathbb{R}^{I_1 \times M}, k$.

We also provide a detailed description of the local block in Algorithm 3. Typically, in standard BTTR, ACE uses the parameters SNR and τ to determine the number of components. Figure 1 shows the breakdown of BTTR into its subfunctions.

In FBTTR, we extend this by introducing a "federated" version of ACE, where the parameters SNR and τ are optimized across multiple institutions. FBTTR needs to make sure that the dimensions of the tensor factorization is equivalent across all clients. Thus, an additional test is performed to ensure that the dimensions of the factor matrices are the same across all clients. If the dimensions are not the same, the server will send the correct dimensions to the clients.

In the end, the server contains the final model, which can be used for prediction. The server can also send the model back to the clients. This way, the clients can use the global model for prediction on their local data. The server can also send the model to new clients, who can use the model for prediction on their data. This way, the model can be used for prediction on new data.



Fig. 1. Hereby we can see the breakdown of BTTR into its subfunctions as described in [6].

Input: num_federated_blocks B

for $k = 1$ to B **do**
  for $k \leftarrow 1 \ldots K$**(in parallel) do**
    $\underline{\mathbf{G}}_{kb}^{(X)}, \mathbf{P}_{kb}^{(2)}, ..., \mathbf{P}_{kb}^{(N)}, SNR, \tau$ = $ACE(\underline{\mathbf{E}}_0, \mathbf{F}_0)$
  end for
  Find SNR_k and τ_k for client k such that all clients have the same dimensions
  for $k \leftarrow 1 \ldots K$**(in parallel) do**
    Send SNR_k and τ_k to client k
    Receive model updates $\underline{\mathbf{G}}_{kb}^{(X)}, \mathbf{P}_{kb}^{(2)}, ..., \mathbf{P}_{kb}^{(N)}$ called $(\Delta W_k^{(t-1)})$ from client's local training (see FBTTR Local Block).
  end for
  $W^{(t)} \leftarrow \frac{1}{\sum_k N_k} \sum_k (N_k \Delta W_k^{(t-1)})$
  for $k \leftarrow 1 \ldots K$**(in parallel) do**
    Send $W^{(t-1)}$ to client k and update the previous block $(\underline{\mathbf{E}}_k, \mathbf{F}_k)$
  end for
end for

return $W(t)$

Fig. 2. FBTTR server with FedAvg aggregation

We can also see that the main distinct factor between FBTTR and other BTTR variations is the seperation of ACE and BTTR. Instead, first, ACE is used locally on each client and the server decides the number of components based on the local ACE results. This way, the server can ensure that the number of components is the same across all clients. This is important because the dimensions of the factor matrices need to be the same across all clients. If the dimensions are not the same, the server will send the correct dimensions to the clients. This way, the server can ensure that the dimensions of the factor matrices are the same across all clients. This also means that FBTTR incorporates horizontal federated learning, as the features are the same across all clients.

B. Automatic Component Extraction (ACE)

Given an N-way variable $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ and a vectorial variable $\mathbf{Y} \in \mathbb{R}^{I_1 \times M}$, we aim to automatically extract the

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 29,2026 at 07:33:18 UTC from IEEE Xplore. Restrictions apply.

Input: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times ... \times I_N}, \mathbf{Y} \in \mathbb{R}^{I_1 \times M}, k$
Output: $\{\mathbf{P}_k^{(n)}\}, \{\mathbf{t}_k\}, \{\mathbf{q}_k\}, \underline{\mathbf{G}}_k^{(X)}$ for $n = 2, ..., N$
if $\left\|\underline{\mathbf{E_k}}\right\| > \epsilon$ and $\|\mathbf{F_k}\| > \epsilon$ **then**
  $\underline{\mathbf{C_k}} = \langle \underline{\mathbf{E}}_k, \mathbf{F}_k \rangle_{\{1,1\}}$
  $\underline{\mathbf{G}}_k, \mathbf{q}_k, \{\mathbf{P}_k^{(n)}\}_{n=2}^N$ = *F-mPSTD*($\underline{\mathbf{E}}_k$, $\mathbf{F}_k$, *SNR**, τ^*)
  $\mathbf{t}_k = (\underline{\mathbf{E}}_k \times_2 \mathbf{P}_k^{(2)T} \times_3 ... \times_N \mathbf{P}_k^{(N)T})_{(1)} vec(\underline{\mathbf{G}}_k^{(X)})$
  $\mathbf{t}_k = \mathbf{t}_k / \|\mathbf{t}_k\|_F$
  $\underline{\mathbf{G}}_k^{(X)} = [\![\underline{\mathbf{E}}_k; \mathbf{t}_k^T, \mathbf{P}_k^{(2)T}, ..., \mathbf{P}_k^{(N)T}]\!]$
  $\mathbf{u_k} = \mathbf{F_k q_k}$
  $\mathbf{d_k} = \mathbf{u_k}^T \mathbf{t_k}$
  {Deflation:}
  $\underline{\mathbf{E_{k+1}}} = \underline{\mathbf{E_k}} - [\![\underline{\mathbf{G}}_k^{(X)}; \mathbf{t}_k, \mathbf{P}_k^{(2)}, ..., \mathbf{P}_k^{(N)}]\!]$
  $\mathbf{F_{k+1}} = \mathbf{F_k} - \mathbf{d_k t_k q_k}^T$
else
  continue
end if

Fig. 3. FBTTR Local Block

Input: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times ... \times I_N}, \mathbf{Y} \in \mathbb{R}^{I_1 \times M}$
Output: $\underline{\mathbf{G}}^{(X)} \in \mathbb{R}^{1 \times R_2 \times ... \times R_N}, \mathbf{q}, \mathbf{t}, \{\mathbf{P}^{(n)}\}_{n=2}^N$
$\underline{\mathbf{C}} = \langle \underline{\mathbf{X}}, \mathbf{Y} \rangle_{(1)}$
Initialisation of $\tau = 90, ..., 100$; *SNR*$= 1, ..., 50$
for *SNR*$_i$ in *SNR* **do**
  for τ_j in τ **do**
    $\underline{\mathbf{G}}, \mathbf{q}, \{\mathbf{P}^{(n)}\}_{n=2}^N$ = *F-mPSTD*($\underline{\mathbf{X}}$, $\mathbf{Y}$, *SNR*$_i$, τ_j)
    calculate BIC value corresponding to *SNR*$_i$ and τ_j using Eq 1
  end for
  select $\tau^* = \text{argmin}_\tau \text{BIC}(\tau)$
  calculate BIC value corresponding to *SNR*$_i$ and τ^* using Eq 1
end for
select *SNR** = argmin_{SNR} BIC(*SNR*, τ^*)
$\underline{\mathbf{G}}, \mathbf{q}, \{\mathbf{P}^{(n)}\}_{n=2}^N$ = *F-mPSTD*($\underline{\mathbf{X}}$, $\mathbf{Y}$, *SNR**, τ^*)
$\mathbf{t} = (\underline{\mathbf{X}} \times_2 \mathbf{P}^{(2)T} \times_3 ... \times_N \mathbf{P}^{(N)T})_{(1)} vec(\underline{\mathbf{G}})$
$\mathbf{t} = \mathbf{t} / \|\mathbf{t}\|_F$
$\underline{\mathbf{G}}^{(X)} = [\![\underline{\mathbf{X}}; \mathbf{t}^T, \mathbf{P}^{(2)T}, ..., \mathbf{P}^{(N)T}]\!]$
return $\underline{\mathbf{G}}^{(X)}, \mathbf{q}, \mathbf{t}, \{\mathbf{P}^{(n)}\}_{n=2}^N$, *SNR**, τ^*

Fig. 4. ACE

latent components $\mathbf{t}$, $\mathbf{q}$ and $P^{(n)}{}_{(n=2)}^N$, associated with the n-th mode of $\underline{\mathbf{X}}$ and maximally correlated with $\mathbf{Y}$, while $\left\|\underline{\mathbf{X}} - [\![\underline{\mathbf{G}}; \mathbf{t}, \mathbf{P}^{(2)}, ..., \mathbf{P}^{(N)}]\!]\right\|_F$ is minimized.

Within ACE, we define the mode-1 cross-product between predictor and response variables as $\underline{\mathbf{C}} = \langle \underline{\mathbf{X}}, \mathbf{Y} \rangle_{(1)}$ and its decomposition as $\underline{\mathbf{C}} \approx [\![\underline{\mathbf{G}}^{(c)}; \mathbf{q}, \mathbf{P}^{(2)}, ..., \mathbf{P}^{(N)}]\!]$. We provide the model with automatic SNR and τ selection based on Bayesian Information Criterion (BIC) defined here as:

Input: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times ... \times I_N}, \mathbf{Y} \in \mathbb{R}^{I_1 \times M}, \tau$,*SNR*
Output: $\underline{\mathbf{G}} \in \mathbb{R}^{1 \times R_2 \times ... \times R_N}, \mathbf{q}, \{\mathbf{P}^{(n)}\}_{n=2}^N$
Initialisation :
$\underline{\mathbf{C}} = \langle \underline{\mathbf{X}}, \mathbf{y} \rangle_{(1)} \in \mathbb{R}^{1 \times I_2 \times ... \times I_N}$
Initialisation of $\{\mathbf{P}^{(n)}\}_{n=2}^N$, $\mathbf{q}$ and $\underline{\mathbf{G}}$ using HOOI on $\underline{\mathbf{C}}$
LOOP Process
repeat
  update $\underline{\mathbf{G}}$ using *SNR*
  prune $\{\mathbf{P}^{(n)}\}_{n=2}^N$, $\mathbf{q}$ and $\underline{\mathbf{G}}$ using τ
until convergence is reached
return $\underline{\mathbf{G}}, \mathbf{q}, \{\mathbf{P}^{(n)}\}_{n=2}^N$

Fig. 5. F-mPSTD

$$BIC(\tau, \text{SNR}|\text{SNR}, \tau^*) = \log\left(\frac{\left\|\underline{\mathbf{C}} - [\![\underline{\mathbf{G}}^{(c)}; \mathbf{q}, \mathbf{P}^{(2)}, ..., \mathbf{P}^{(N)}]\!]\right\|_F}{s}\right) + \frac{\log(s)}{s} DF, \quad (1)$$

where $\underline{\mathbf{G}}^{(c)}$, q and $\{\mathbf{P}^{(n)}\}_{n=2}^N$ are the sparse core, latent vector and factor matrices obtained with F-mPSTD [6] - a modified version of the original sparse Tucker decomposition (PSTD) [23] - using specific τ and SNR values, s the number of entries in $\underline{\mathbf{G}}$, and DF the degree of freedom calculated as the number of non-zero elements in $\underline{\mathbf{G}}^{(c)}$, as suggested in [24]. For each SNR value, the associated optimal τ is computed as $\tau^* = \text{argmin}_\tau BIC(\tau, \text{SNR})$. Then, the optimal SNR is determined as $\text{SNR}^* = \text{argmin}_{\text{SNR}} BIC(\text{SNR}, \tau^*)$. Once $\underline{\mathbf{G}}^{(c)}$, q and $\{\mathbf{P}^{(n)}\}_{n=2}^N$ are computed, the score vector t is first calculated as

$$\mathbf{t} = (\underline{\mathbf{C}} \times_2 \mathbf{P}^{(2)T} \times_3 ... \times_N \mathbf{P}^{(n)T})_{(1)} \text{vec}(\underline{\mathbf{G}}^{(c)}),$$

and then normalized. This is summarized in Algorithm 4.

The F-mPSTD model is first initialized with higher-order orthogonal iteration (HOOI) [25]. Then, iteratively, a soft-thresholding rule based on parameter λ, alternated with a threshold τ, are applied to enhance model sparsity and to prune irrelevant components, respectively. Note that in [23] SNR $\in [1, 50]$ is used to derive, via a line search, the optimal degree of sparsity λ of the core tensor (see [23]. At each iteration, the core tensor $\underline{\mathbf{G}}$ is updated using the soft-thresholding rule as $\underline{\mathbf{G}} = sgn(\underline{\mathbf{G}}) \times max\{|\underline{\mathbf{G}}| - \lambda, 0\}$, while the threshold $\tau \in [0, 100]$ is used to reject unnecessary components from the n-mode $S^{(n)} = \{r | 100(1 - \frac{\sum_i \mathbf{G}_{(n)(r,i)}}{\sum_{t,i} \mathbf{G}_{(n)(t,i)}}) \geq \tau\}$, $\mathbf{P}^{(n)} = \mathbf{P}^{(n)}(:, S^{(n)})$, $\mathbf{q} = \mathbf{q}(S^{(n)})$ and $\mathbf{G}^{(n)} = \mathbf{G}^{(n)}(S^{(n)}, :)$. The full mathematical breakdown can be found in [23]. The F-mPSTD is summarized in Algorithm 5.

IV. Results and Discussion

A. Case Study: Brain-Computer Interfacing Finger Movement Decoding

We will compare FBTTR's performance with that of the aforementioned non-multilinear models, as well as with multi-

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 29,2026 at 07:33:18 UTC from IEEE Xplore. Restrictions apply.

linear models HOPLS and BTTR, in predicting continuous finger flexions from ECoG recordings. We will use the publicly-available BCI Competition IV dataset 4 for reproducibility's sake. It comprises ECoG signals sampled at 1000 Hz from the motor cortex (hand-knob area) of three subjects, as well as the time courses of the flexion of each of five fingers of the contralateral hand (measured with a data glove). There are 150 trials (samples) in total (30 trials per finger) recorded in a single session (600 seconds). Subjects flexed the cued finger 3-5 times for 2 seconds followed by a rest period of 2 seconds. The first 400 and the last 200 seconds of recording were used as training and testing sets, respectively. The number of ECoG electrodes (channels) was 62 for Subject 1, 48 for Subject 2, and 64 for Subject 3. More details about the data can be found in [26].

The dataglove signal lagged the amplifier by 37 ms ($\pm$3 ms, SEM), consistent with its 25 Hz sampling interval (40 ms). To compensate, glove data were shifted by one sample, as in previous work.

For each trial, glove and ECoG data were extracted from 1 s before onset. ECoG preprocessing included notch filtering at 50/100 Hz, removal of bad channels (55 in Subject 1; 21, 38 in Subject 2; 50 in Subject 3), and re-referencing with a Common Average Reference (CAR) [27]. Signals were transformed into a 4th-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{\text{Samples} \times \text{Channels} \times \text{Frequencies} \times \text{Time}}$, where Samples correspond to glove trajectory length, Channels to curated electrodes (61, 46, 63), Frequencies to eight bands from 1.5–130 Hz via Butterworth filters [28], and Time to 10 bins from a 1 s window downsampled to 10 Hz. Glove trajectories were z-scored per finger.

1) Parameter optimization multilinear models, performance assessment: In order to optimize the model parameters from the training data, $K, R_2, ..., R_N$ for HOPLS and K for BTTR, eBTTR and FBTTR, a 5-fold cross-validation approach was used. Since the cited BCI Competition IV dataset 4 studies reported Pearson's correlation coefficients between data glove- and predicted intended finger trajectories, we also applied it here. In support of the statistical analysis, the test data was split in 5 non-overlapping blocks. We used the two-tailed Wilcoxon signed-rank test [29] to compare average accuracies per finger and subject; they are considered significantly different if the p-value is < 0.05.

2) Results and Impact: The Pearson correlation coefficients are reported for the intended movements of each finger individually, and averaged across all fingers except for finger 4 (ring) as flexing the latter is difficult to suppress when the 3rd or 5th finger is flexing, as this was done by the authors of the cited papers. The average results and their standard deviations for the 5 blocks of test data (see above) are listed in Table II, Table III and Table IV (listed under Avg.) for subjects 1, 2 and 3, respectively.

A statistically significant difference between FBTTR and HOPLS for subjects 1 and 2, mainly due to the difference in correlation coefficients of the middle finger. Interestingly, there is no significant performance difference between FBTTR and eBTTR for subject 1 and subject 2, but there is for subject 3. This is due to the difference in correlation coefficients of the thumb. Thus, overall, we can say there is a small, but significant improvement of using FBTTR over eBTTR.

Looking at the model parameters for subject 1 (Figure 6), we can see that there is generally a smooth transition from block to block for each finger and thus, the transition between these blocks is quite stable over time. This tells us that the model is learning the underlying structure of the data well. It also tells us that the FedAVG aggregation method is working well, as the model parameters are quite stable over time and not causing any significant fluctuations.

We also investigated the runtime needed to estimate the model parameters. One of the main advantages of BTTR is its speed. BTTR is trained significantly faster than HOPLS (e.g. 3 minutes against 14 hours for HOPLS). Note that HOPLS requires computationally expensive techniques such as cross-validation to identify the optimal set of model parameters (i.e., the number of scores and loadings). Since FBTTR is based on BTTR, it enjoys the latter's fast training time (e.g. 5 minutes against 3 minutes for BTTR). Averaging the model parameters doesn't impact the runtime significantly. However, there are likely some performance improvements that can be made. Generally, it's not required to train the model from scratch for each new client. Instead, the model can be fine-tuned using a few epochs of training on the client's data. This is a common practice in federated learning, where the global model is fine-tuned on the client's data to adapt to the local data distribution. This fine-tuning process is much faster than training the model from scratch. Interestingly enough, there's little communication overhead or memory costs. The memory costs of several smaller clients is much smaller compared to a single large centralised model.

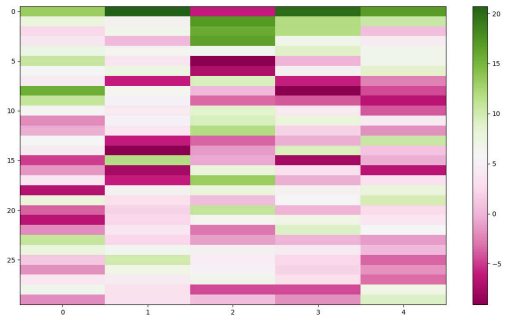


Fig. 6. Model parameters of Subject 1 for the first 30 blocks. The x-axis represents the fingers (Thumb, Index, Middle, Ring and Pinky respectively), while the y-axis represents the various blocks in order. The color intensity represents the strength of the connection in terms of Pearson correlation.

B. Case Study: Heart Disease & Cancer Risk Prediction

Our dataset for this case study comprises two real-world clinical datasets from Flamby [30]: Fed-Heart-Disease and Fed-Tcga-Brca. The Fed-Heart-Disease dataset contains 740 records from four centers, detailing 13 clinical features and

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 29,2026 at 07:33:18 UTC from IEEE Xplore. Restrictions apply.

Methods	Thumb	Index	Middle	Ring	Pinky	Avg.
F-BTTR	0.72 ± .06	0.77 ± .09	0.45 ± .02	0.68 ± .04	0.67 ± .02	0.64 ± .05
eBTTR	0.71 ± .04	0.75 ± .06	0.49 ± .06	0.69 ± .04	0.68 ± .01	0.66 ± .03
HOPLS	0.70 ± .05	0.79 ± .08	**0.36 ± .03**	0.70 ± .06	0.65 ± .02	0.63 ± .04
AM	**0.57 ± .03**	0.69 ± .06	**0.14 ± .02**	**0.52 ± .04**	**0.28 ± .01**	**0.42 ± .03**
RF	**0.58 ± .09**	**0.54 ± .05**	**0.07 ± .03**	**0.31 ± .05**	**0.33 ± .02**	**0.38 ± .05**
LARS	**0.11 ± .05**	**0.08 ± .03**	**0.10 ± .02**	0.60 ± .05	**0.39 ± .02**	**0.17 ± .03**
CNN	**0.67 ± .04**	0.78 ± .04	**0.11 ± .02**	**0.54 ± .03**	**0.45 ± .04**	**0.50 ± .04**
LSTM	0.73 ± .03	0.79 ± .08	**0.18 ± .02**	0.61 ± .04	**0.45 ± .04**	**0.54 ± .04**

TABLE II
INTENDED (CUED) FINGER MOVEMENT ACCURACY FOR SUBJECT 1 (PEARSON CORRELATION).

Methods	Thumb	Index	Middle	Ring	Pinky	Avg.
F-BTTR	0.66 ± .06	0.47 ± .09	0.32 ± .02	0.52 ± .04	0.48 ± .02	0.48 ± .05
eBTTR	0.63 ± .05	0.47 ± .08	0.33 ± .03	0.52 ± .02	0.47 ± .01	0.48 ± .05
HOPLS	0.63 ± .04	0.47 ± .06	**0.26 ± .05**	0.51 ± .02	0.48 ± .01	0.44 ± .04
AM	**0.52 ± .03**	**0.36 ± .06**	**0.23 ± .02**	0.48 ± .04	**0.33 ± .01**	**0.36 ± .03**
RF	**0.52 ± .05**	**0.36 ± .04**	**0.22 ± .03**	**0.39 ± .04**	**0.25 ± .02**	**0.34 ± .04**
LARS	**0.54 ± .05**	0.41 ± .04	**0.18 ± .02**	**0.44 ± .04**	**0.25 ± .02**	**0.35 ± .03**
CNN	0.60 ± .04	0.40 ± .04	**0.24 ± .02**	**0.44 ± .03**	**0.28 ± .04**	**0.38 ± .04**
LSTM	0.62 ± .03	0.38 ± .08	0.27 ± .02	0.47 ± .04	**0.30 ± .04**	**0.39 ± .04**

TABLE III
IDEM TO TABLE II BUT FOR SUBJECT 2.

a binary heart disease indicator for each patient. We used a logistic regression model for prediction. The Fed-Tcga-Brca dataset, sourced from The Cancer Genome Atlas's Genomics Data Commons portal, has data from 1,088 breast cancer patients across six centers. With 39 clinical features and each patient's time of death, we used a Cox model to predict the risk of death. All non-BTTR results use either logistic regression or a Cox model for the Fed-Heart-Disease and Fed-Tcga-Brca respectively.

Our experimental setup involves training the FBTTR model on federated data partitions. We evaluate the model comparing its performance against centralized BTTR and other baseline models using AUC and accuracy. To illustrate the effectiveness of FBTTR, we follow different experimental scenarions as described in the FL4E framework [31]. Foundational scripts that can be generalized and act as guidelines for hybrid experiment settings in the FL4E framework were used for these experiments.

- **Fully Federated Experiment**: All participating clients contribute to the global model using their local datasets, adhering to the conventional FL setup.
- **Hybrid Experiment**: Some clients contribute their data centrally while others participate as federated clients. This scenario demonstrates how partial centralization can influence model performance.
- **Centralized Experiment**: All clients send their data to a central server for model training. This serves as a benchmark to compare the federated and hybrid approaches against a fully centralized model.

Methods	Thumb	Index	Middle	Ring	Pinky	Avg.
FBTTR	0.76 ± .05	0.57 ± .06	0.64 ± .02	0.62 ± .02	0.76 ± .01	0.68 ± .05
eBTTR	**0.71 ± .05**	0.57 ± .07	0.64 ± .04	0.62 ± .02	0.73 ± .01	0.66 ± .05
HOPLS	0.74 ± .06	0.57 ± .09	0.65 ± .02	0.61 ± .04	**0.68 ± .02**	0.64 ± .04
AM	**0.59 ± .03**	0.51 ± .06	**0.32 ± .02**	**0.53 ± .04**	**0.42 ± .01**	**0.46 ± .03**
RF	0.67 ± .05	**0.27 ± .04**	**0.16 ± .03**	**0.14 ± .04**	**0.36 ± .02**	**0.37 ± .04**
LARS	0.72 ± .05	**0.43 ± .04**	**0.45 ± .02**	**0.51 ± .04**	**0.64 ± .02**	**0.56 ± .03**
CNN	0.74 ± .03	0.53 ± .05	**0.45 ± .04**	**0.49 ± .03**	0.68 ± .06	**0.60 ± .05**
LSTM	0.74 ± .02	0.55 ± .06	**0.46 ± .04**	**0.41 ± .02**	0.75 ± .06	**0.62 ± .05**

TABLE IV
IDEM TO TABLE II BUT FOR SUBJECT 3.

Although FBTTR, for now, only implements the equivalent for the FedAVG algorithm, we still made the comparison with other federated learning strategies, such as FedOpt and FedProx, to evaluate its performance in a federated setting [32], [33].

1) Parameter optimization multilinear models, performance assessment: In order to optimize the model parameters from the training data, K for BTTR, eBTTR and FBTTR, a 5-fold cross-validation approach was used. Hyperparameter tuning for standard federated learning was identified in the previous study [31]. This was performed using grid search on a centralized setting and applied to each experiment of the data sets. To ensure consistency, each experiment was repeated five times.

While Pearson Correlation's are the standard for finger movement decoding, this dataset is evaluated using Area Under the Receiver Operating Characteristic Curve (ROC-AUC) and accuracy metrics for the Fed-Heart-Disease dataset and the concordance index (C-index) for the Fed-Tcga-Brca dataset. These metrics were chosen according to their ability to assess model performance in binary classification and survival analysis task, while accounting for potential class imbalance. Similar to the other case study, in support of the statistical analysis, the test data was split in 5 non-overlapping blocks. We used the two-tailed Wilcoxon signed-rank test [29] to compare average accuracies; they are considered significantly different if the p-value is < 0.05.

2) Results and Impact: The case study results highlight the practical applicability of FBTTR in predicting heart disease. The implementation demonstrated that FBTTR could be a viable solution for predictive analytics in healthcare. Similarly as in the other case study, we found that federated models exhibit similar to superior performance compared to centralized models. This is conform to results found in the literature [31]. The hybrid experiment also showed promising results, indicating that a combination of centralized and federated learning can be beneficial in certain scenarios and can be a potential avenue for FBTTR.

The results for the Fed-Heart-Disease Dataset are shown in Table VI. A first observation is that FBTTR outperforms all other federated learning strategies, including FedAvg, FedAdagrad, FedYogi, and FedProx, as well as the hybrid and centralized experiments, in absolute numbers, with most comparisons being statistically significant.

As seen by Table V, it can be seen that there was a large class imbalance introduced in the various clients. The same pattern as found in the FL4E study can be observed here and showcase the effects of class imbalance and quantity skew, suggesting that smaller clients with imbalanced data sets may benefit more from participating in an FL environment than operating in isolation. This insight is harmonious with the foundational principles of the FL paradigm, which aspires to harness the potential of diverse data sources to enhance model robustness and generalizability. It should be noted that the larger clients can showcase a decrease of performance due to participating in an FL environment.

While there is a slight accuracy drop in the federated

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 29,2026 at 07:33:18 UTC from IEEE Xplore. Restrictions apply.

TABLE V
DELINEATING CLIENT HETEROGENEITY AND CLASS IMBALANCE IN THE FED-HEART-DISEASE ANALYSIS.

Client	Class 0	Class 1
Client 0	164	139
Client 1	163	98
Client 2	1	45
Client 3	29	101

setting compared to the centralized setting, the difference is not statistically significant, and there is practically no difference in the ROC-AUC.

The hybrid experiment for FBTTR highlights an important finding: it delivers performance on par with fully federated models, while significantly outperforming local setups for smaller clients. At the same time, it avoids much of the complexity and overhead of coordinating federation across multiple data centers. This approach allows centralized management of in-house data, giving analysts direct access (within regulatory limits) and the flexibility to perform advanced analyses. With this access, analysts can design more effective models and explore complex questions with greater freedom.

The results for the Fed-Tcga-Brca Dataset are shown in Table VII. Here, FBTTR also outperforms all other federated learning strategies, including FedAvg, FedAdagrad, FedYogi, and FedProx, as well as the hybrid and centralized experiments, in absolute numbers, with most comparisons being statistically significant. The only comparisons that are not significant are between FBTTR and FedAdagrad, and FBTTR and FedProx. A possible explanation for this is that the FedAdagrad and FedProx algorithms are more robust to the noise in the data, which could explain why they perform better compared to the other federated learning strategies in this case.

FBTTR also performs statistically significantly better than BTTR in the Fed-TCGA-BRCA Datasets. This is an important result, as it shows that FBTTR is able to leverage the federated data to improve the prediction performance compared to a centralized model.

V. CONCLUSION

In the first case study, using the BCI Competition IV dataset, we compared FBTTR with non-multilinear models for finger movement decoding. Our results show that FBTTR outperforms the other models, demonstrating its effectiveness in predicting finger movements from ECoG signals. It performs on par, and in some cases, better compared to centralised versions of BTTR.

In the second case study, we applied FBTTR to predict heart disease using two real-world clinical datasets. Our findings indicate that FBTTR performs well in this domain, showcasing its potential for healthcare analytics other than finger movement decoding. The second usecase also shows the superior performance of FBTTR compared to standard federated learning, as well as the equal or superior performance compared to BTTR. Overall, our results suggest that FBTTR is a versatile and powerful tool for a wide range of applications, from brain-computer interfacing to predictive healthcare analytics.

FBTTR represents a significant advancement in applying tensor regression in federated learning environments. The method's ability to handle high-dimensional data while in a decentralised setting makes it particularly suited for healthcare applications. However, there are limitations, such as the computational overhead associated with model synchronization and the potential for performance degradation with highly

TABLE VI
EXPERIMENTS USING FL4E ([31]) AND FBTTR WITH THE FED-HEART-DISEASE DATASET. SIGNIFICANTLY DIFFERENT RESULTS COMPARED FBTTR ARE INDICATED IN BOLD.

Method	ROC-AUC	Accuracy
(Fully Federated) FBTTR	0.872 ± 0.02	0.772 ± 0.02
(Centralised) BTTR	0.874 ± 0.03	0.783 ± 0.04
Hybrid Experiment FBTTR	0.872 ± 0.02	0.772 ± 0.02
Local Experiment FBTTR		
Client 0	0.89 ± 0.01	0.81 ± 0.01
Client 1	0.92 ± 0.01	0.84 ± 0.01
Client 2	0.64 ± 0.31	**0.63 ± 0.21**
Client 3	0.68 ± 0.22	**0.72 ± 0.2**
Fully Federated		
FedAvg	0.846 ± 0.002	0.733 ± 0.007
FedAdagrad	0.841 ± 0.029	**0.726 ± 0.05**
FedYogi	0.803 ± 0.051	**0.715 ± 0.032**
FedProx	0.846 ± 0.003	0.741 ± 0.006
Hybrid Experiment		
FedAvg	0.825 ± 0.004	0.740 ± 0.054
FedAdagrad	0.821 ± 0.008	0.741 ± 0.005
FedYogi	0.794 ± 0.013	**0.710 ± 0.039**
FedProx	0.822 ± 0.004	**0.737 ± 0.012**
Local Experiment		
Client 0	0.842 ± 0.009	0.753 ± 0.011
Client 1	0.882 ± 0.007	0.800 ± 0.013
Client 2	0.546 ± 0.271	**0.550 ± 0.199**
Client 3	0.542 ± 0.054	**0.559 ± 0.096**
Centralized	0.812 ± 0.003	0.753 ± 0.007

TABLE VII
EXPERIMENTS USING FL4E ([31]) AND FBTTR WITH THE FED-TCGA-BRCA DATASET. SIGNIFICANTLY DIFFERENT RESULTS COMPARED FBTTR ARE INDICATED IN BOLD.

Method	C-Index
FBTTR	0.775 ± 0.01
BTTR	**0.737 ± 0.02**
Fully Federated	
FedAvg	**0.732 ± 0.030**
FedAdagrad	0.748 ± 0.016
FedYogi	0.745 ± 0.037
FedProx	**0.725 ± 0.007**
Hybrid Experiment	
FedAvg	**0.656 ± 0.06**
FedAdagrad	0.776 ± 0.036
FedYogi	**0.726 ± 0.041**
FedProx	**0.439 ± 0.227**
Local Experiment	
Client 0	**0.668 ± 0.064**
Client 1	**0.445 ± 0.237**
Client 2	**0.635 ± 0.166**
Client 3	**0.570 ± 0.140**
Client 4	**0.851 ± 0.078**
Client 5	**0.666 ± 0.001**
Centralized	**0.609 ± 0.207**

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 29,2026 at 07:33:18 UTC from IEEE Xplore. Restrictions apply.

heterogeneous data. Future research could explore optimizing the aggregation process and extending the method to other medical conditions. In this work, there was also no specific focus placed on any privacy-preserving mechanism. Due to the sensitive nature of medical data, this would be an interesting avenue to explore.

In conclusion, this paper introduced Federated Block-Term Tensor Regression (FBTTR), an innovative approach that extends BTTR to federated learning settings. Our findings demonstrate that FBTTR effectively predicts heart disease, making it a valuable tool for healthcare analytics. Future work will focus on refining the model and exploring its application in other domains.

REFERENCES

[1] Qibin Zhao, Cesar F Caiafa, Danilo P Mandic, Zenas C Chao, Yasuo Nagasaka, Naotaka Fujii, Liqing Zhang, and Andrzej Cichocki. Higher order partial least squares (hopls): a generalized multilinear regression method. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1660–1673, 2012.

[2] Andrey Eliseyev and Tetiana Aksenova. Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording. *PloS one*, 11(5):e0154878, 2016.

[3] Alim Louis Benabid, Thomas Costecalde, Andrey Eliseyev, Guillaume Charvet, Alexandre Verney, Serpil Karakas, Michael Foerster, Aurélien Lambert, Boris Morinière, Neil Abroug, et al. An exoskeleton controlled by an epidural wireless brain–machine interface in a tetraplegic patient: a proof-of-concept demonstration. *The Lancet Neurology*, 18(12):1112–1122, 2019.

[4] Henri Lorach, Andrea Galvez, Valeria Spagnolo, Felix Martel, Serpil Karakas, Nadine Intering, Molywan Vat, Olivier Faivre, Cathal Harte, Salif Komi, et al. Walking naturally after spinal cord injury using a brain–spine interface. *Nature*, pages 1–8, 2023.

[5] Flavio Camarrone, Mariana P Branco, Nick F Ramsey, and Marc M Van Hulle. Accurate offline asynchronous detection of individual finger movement from intracranial brain signals using a novel multiway approach. *IEEE Transactions on Biomedical Engineering*, 68(7):2176–2187, 2020.

[6] Axel Faes, Flavio Camarrone, and Marc M Van Hulle. Single finger trajectory prediction from intracranial brain activity using block-term tensor regression with fast and automatic component extraction. *IEEE Transactions on Neural Networks and Learning Systems*, submitted 2022.

[7] Gang Pan, Jia-Jun Li, Yu Qi, Hang Yu, Jun-Ming Zhu, Xiao-Xiang Zheng, Yue-Ming Wang, and Shao-Min Zhang. Rapid decoding of hand gestures in electrocorticography using recurrent neural networks. *Frontiers in neuroscience*, 12:555, 2018.

[8] A Faes and MM Van Hulle. Finger movement and coactivation predicted from intracranial brain activity using extended block-term tensor regression. *Journal of Neural Engineering*, 19(6):066011, 2022.

[9] Bingyan Liu, Nuoyan Lv, Yuanchun Guo, and Yawen Li. Recent advances on federated learning: A systematic survey. *Neurocomputing*, page 128019, 2024.

[10] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2017.

[11] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, David Huba, Aleksandr Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Steven Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. In *Proceedings of the 2nd SysML Conference*, 2019.

[12] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. In *NIPS Workshop on Privacy and Security in Machine Learning*, volume 30, 2017.

[13] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[14] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yue Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11, 2019.

[15] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[16] Qiang Yang, Yang Liu, Tianjian Cheng, and Yan Kang. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[17] Andrew Hard, Kanishka Rao, Robert Mathews, Satyen Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloe Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[18] Micah J Sheller, Brandon Edwards, Guido A Reina, James Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Wei Xu, Daniel Marcus, and Spyridon Bakas. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports*, 10(1):1–12, 2020.

[19] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[20] D. Nguyen Duy, M. Affenzeller, and R. Nikzad-Langerodi. Towards vertical privacy-preserving symbolic regression via secure multiparty computation. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, pages 2420–2428, 2023.

[21] Nanying Liang and Laurent Bougrain. Decoding finger flexion from band-specific ecog signals in humans. *Frontiers in neuroscience*, 6:91, 2012.

[22] Ziqian Xie, Odelia Schwartz, and Abhishek Prasad. Decoding of finger trajectory from ecog using deep learning. *Journal of neural engineering*, 15(3):036009, 2018.

[23] Tatsuya Yokota and Andrzej Cichocki. Multilinear tensor rank estimation via sparse tucker decomposition. In *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, pages 478–483. IEEE, 2014.

[24] Genevera I Allen and Mirjana Maletić-Savatić. Sparse non-negative generalized pca with applications to metabolomics. *Bioinformatics*, 27(21):3029–3035, 2011.

[25] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. *SIAM journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

[26] Kai J Miller and Gerwin Schalk. Prediction of finger flexion: 4th brain-computer interface data competition. *BCI Competition IV*, 1:1–2, 2008.

[27] Collin D Binnie. *Clinical neurophysiology: EEG, paediatric neurophysiology, special techniques and applications*, volume 2. Elsevier Health Sciences, 2003.

[28] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[29] Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer, 1992.

[30] Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, et al. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. *Advances in Neural Information Processing Systems*, 35:5315–5334, 2022.

[31] Ashkan Pirmani, Martijn Oldenhof, Liesbet M Peeters, Edward De Brouwer, and Yves Moreau. Accessible ecosystem for clinical research (federated learning for everyone): Development and usability study. *JMIR Formative Research*, 8:e55496, 2024.

[32] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[33] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 29,2026 at 07:33:18 UTC from IEEE Xplore. Restrictions apply.