Multiarea Inertia Estimation Using Convolutional Neural Networks and Federated Learning

Abodh Poudyal, *Student Member, IEEE*, Ujjwol Tamrakar, *Member, IEEE*,
Rodrigo D. Trevizan, *Member, IEEE*, Robert Fourney, *Member, IEEE*,
Reinaldo Tonkoski, *Senior Member, IEEE*, and Timothy M. Hansen, *Senior Member, IEEE*

***Abstract*—With the increase in penetration of renewable energy sources (RES), traditional inertia estimation techniques based purely on the number of online synchronous generators are increasingly unsuitable, ultimately leading towards suboptimal frequency control in the electric power grid. The stochastic nature of RES additionally makes the system inertia a time-varying quantity. Furthermore, the frequency and inertial response of power systems change drastically in multiarea power systems with interconnected tie-lines. Hence, it is important for state/parameter estimation (e.g., inertia) in multiarea systems, while ensuring communication between each of the areas. In this article, a client–server-based federated learning framework is used to estimate power system inertia in a multiarea system. Federated learning is a machine learning technique where multiple decentralized devices are trained with local data, and a global model is updated and redistributed by a central server by aggregating the trained weights of the decentralized devices, without exchanging the local data. Using local frequency measurements, obtained from the phase-locked loop of an energy storage system, the inertia at each of the areas can be estimated locally via offline training using convolutional neural networks (CNNs), whereas the CNN weights update in an online fashion. The framework, tested on a two-area power system, accurately estimated the inertia constant for both independent and identically distributed (IID) and non-IID data. Furthermore, the CNN-based method outperformed conventional neural network-based estimation techniques in terms of number of communication rounds and estimation accuracy.**

***Index Terms*—Convolutional neural networks (CNNs), federated learning (FL), low-inertia grids, multiarea power system, power system inertia estimation.**

Nomenclature

Abbreviations

ACE	Area control error.
AWGN	Add white Gaussian noise.
CNN	Convolutional neural network.
DER	Distributed energy resources.
ERCOT	Electricity Reliability Council of Texas.
ESS	Energy storage system.
FFR	Fast frequency response.
FL	Federated learning.
GPU	Graphics processing unit.
IID	Independent and identically distributed.
MLP	Multilayer perceptron.
MSE	Mean squared error.
PLL	Phase-locked loop.
PMU	Phasor measurement unit.
RES	Renewable energy source.
RMSE	Root MSE.
ROCOF	Rate-of-change-of-frequency.
RRS	Response reserve service.
SNR	Signal-to-noise ratio.
UFLS	Under frequency load shedding.

Machine Learning Variables and Parameters

α	Learning rate.
β	Local mini-batch input, $\beta \in \mathbb{B}$.
$\mathbb{B}$	Set of local mini-batches of training snapshots, $\mathbb{B} \in \mathbb{P}_n$.
$\mathbb{E}$	Number of local training epochs.
$\mathbb{P}_n$	Number of training snapshots at ESS client *n*.
$\mathcal{R}$	Kernel size for CNN layer 1.
$\mathcal{S}$	Kernel size for CNN layer 2.
μ	Total number of training snapshots at for *N* clients.
μ_n	Number of training snapshots at client n.
∇E_{MSE}	Gradient of MSE loss function.
Ψ_i	Selected ESS clients in communication round *i*.
$\tilde{M}$	Estimated value of inertia constant.

Manuscript received 9 July 2021; revised 14 October 2021; accepted 3 December 2021. Date of publication 29 December 2021; date of current version 9 December 2022. This work was supported in part by the U.S. Department of Energy Office of Science, Office of Basic Energy Sciences, EPSCoR Program, in part by the Office of Electricity Microgrid R&D Program, in part by the Office of Energy Efficiency and Renewable Energy Solar Energy Technology Office under Grant DE-SC0020281, and in part by the South Dakota State University Joint Research, Scholarship, and Creative Activity Challenge Fund. This research made use of the "Roaring Thunder" cluster at South Dakota State University, Funded by National Science Foundation under Grant CNS-01726946 and Grant OAC-1924302. The work of Ujjwol Tamrakar and Rodrigo D. Trevizan at Sandia National Laboratories was supported by the U.S. Department of Energy, Office of Electricity, Energy Storage Program. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy National Nuclear Security Administration under Grant DE-NA-0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. SAND2021-15467J. *(Corresponding author: Abodh Poudyal.)*

Abodh Poudyal is with the Department of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99163 USA (e-mail: abodh.poudyal@wsu.edu).

Robert Fourney, Reinaldo Tonkoski, and Timothy M. Hansen are with the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD 57007 USA (e-mail: robert.fourney@ieee.org; tonkoski@ieee.org; timothy.hansen@sdstate.edu).

Ujjwol Tamrakar and Rodrigo D. Trevizan are with the Sandia National Laboratories, Albuquerque, NM 87185-5820 USA (e-mail: utamrak@sandia.gov; rdtrevi@sandia.gov).

Digital Object Identifier 10.1109/JSYST.2021.3134599

ξ	Power systems area number.
b	Local mini-batch input size.
C	Fraction of clients selected by the server, $C \leq N$.
c	Size of single input vector in a batch.
h_1	Number of neurons in first hidden layer of feedforward network.
h_2	Number of neurons in first hidden layer of feedforward network.
m	Number of selected ESS clients.
N	Number of ESS clients.
p	Number of channels for CNN layer 1.
q	Number of channels for CNN layer 2.
w_0	Initial weight of CNN.
w_t	Aggregated CNN weights.
Y	Input size of feedforward neural network.

Power System Variables and Parameters

$\Delta\omega$	Change in frequency.
$\Delta\dot{\omega}$	Rate-of-change-of-frequency.
ΔP_L	Change in load.
$\Delta\delta$	Change in power angle.
ΔP_e	Change in electrical power.
ΔP_m	Change in mechanical power
ΔP_{tie}	Tie-line power flow.
B	Bias factor.
D	Damping coefficient.
K	Secondary controller gain.
M	Inertia constant.
R	Speed regulation droop.
T_0	Tie-line coefficient.
T_g	Turbine-governor time constant

I. Introduction

INCREASING penetration of renewable energy sources (RESs) results in a decline of power system inertia. Power system inertia slows the initial rate-of-change-of-frequency (ROCOF) in the case of a frequency event [1], [2]. With a significant decline in inertia, the instantaneous ROCOF may be large enough to trigger under frequency load shedding that could initiate a sequence of cascading outages [3]–[5].

As a result of the stochastic nature of RES, power system inertia is also a time-varying parameter in the modern grid [6]–[9]. With reduced inertia in the grid from replacing conventional synchronous generation assets with an increased penetration of RES, it is necessary to perform real-time monitoring of inertia levels in the power grid [10]–[12]. Furthermore, due to the absence of rotating components, inverter-based systems are considered passive in terms of inertial response. However, recent advancements in control techniques led to the provision of virtual inertia support from RES-based systems [13]–[15]. With the meshed architecture of the grid, the increasing penetration of stochastic RES makes the task of inertia estimation even more challenging as inertial support can also come from interconnected areas via tie-lines [16], [17]. Recently, we presented a preliminary inertia estimation method using convolutional neural networks (CNNs) for a single area power system [18]. The estimation of inertia was based purely on frequency and ROCOF measurements taken locally from the phase-locked loop (PLL) of an energy storage system (ESS). In the proposed work, the ESS is first used to probe the power system, and then the PLL of the same ESS is used for inertia estimation. As an extension to the former work, this article includes the effect of tie-line power flow in multiarea power systems while estimating the inertia constant in a decentralized fashion.

A. Literature Review

Several inertia estimation techniques using phasor measurement units (PMUs) were presented in [19]–[23]. Lugnani *et al.* [19] estimated the inertia constant of a single or group of generators with the help of PMUs installed on the generator bus. A low-order autoregressive moving average model was used to estimate the value of inertia. Panda *et al.* [20] proposed an online inertia estimation technique based on synchronized PMU measurements. The dynamics of the network was emulated via an equivalent swing equation in terms of frequency deviation and inertia constant. The Electric Reliability Council of Texas (ERCOT) monitors the grid inertia based on the unit commitment plans of interconnected generators in the grid [12]. However, the nonsynchronous generating units have not yet been incorporated in the tool [24]. Recently, ERCOT promulgated new ancillary market design that allows fast frequency response (FFR) under response reserve service [25]. Hence, the estimation of inertia constant based purely on the online status of synchronous units is inaccurate considering the FFR support that can come from the interconnected nonsynchronous units. There are some model-free techniques to estimate the inertia constant. A Markov–Gaussian-based inertia estimation technique, discussed in [26], can dynamically estimate the system inertia. However, a large historical dataset (i.e., two years) was required for this method, and this may not be readily available for other systems and can be cumbersome for training for moderate sampling time.

The electric power grid is highly meshed, and it is necessary to consider the impact of inter-area power flow while estimating the inertia constant. There is little work that considers multiarea power systems when estimating the inertia constant. Tuttelberg *et al.* [23] estimated the equivalent inertia of individual areas in a multiarea power system using ambient PMU measurements. However, only synchronous units have been considered and, in general, PMU data should be postprocessed to capture the accurate inertial response. An inertia estimation method for a multiarea interconnected electric power system using electromechanical oscillation modes has been discussed in [27]. The relation between inertia, frequency, and damping of electromechanical modes was developed, however the penetration of RES in any one of the areas can drastically change the dynamics of electromechanical modes [28], [29]. Hence, estimating inertia with respect to number of online synchronous units is increasingly inaccurate considering the inertial response coming from RES-based resources.

Schmitt *et al.* [30] used inter-area modal information, particularly frequency and damping of the oscillation mode, to estimate the inertia constant. A general neural network-based inertia estimation technique was presented that estimates the inertia only for system with synchronous generation. Additionally, the training data was centralized and trained in a single neural network. Aggregating the data centrally for training increases the cost and makes this approach less secure. A data-driven inertia estimation approach in multiarea systems is discussed in [31]. The dependency of the system inertia with the eigenvalue and eigenvectors, extracted using dynamic mode decomposition, were analyzed. However, only synchronous units are considered as a part of the analysis.

With limited data-driven techniques in the literature, most above-mentioned inertia estimation methods are based on the number of online synchronous units in the power system. With increasing penetration of RES-based units, it is necessary to include the effects of nonsynchronous units in grid inertia [7], [32], [33]. It is therefore preferable to design a data-driven generator-independent technique that can better identify the uncertain dynamic behavior of modern power grids that have a significant penetration of stochastic RES-based units [34], [35]. As mentioned before, such data-driven estimation techniques will be independent of the generating units used, and can be generalized to estimate the inertia for any system. Furthermore, such methods should be cost efficient, accurate, and computationally effective. The centralized estimation method can be highly inefficient as the overhead incurred during the data collection phase outweighs the purpose of the estimation task. Furthermore, centralized approaches are less secure as compared to decentralized training methods [36].

B. Key Contributions

The conventional inertia estimation methods depend on the types of generating units being used and are mostly centralized. Here, we propose a framework to estimate the inertia of the system in a decentralized fashion that depends only on the local frequency measurements. Recently, a client–server-based framework—federated learning (FL)—has gained popularity as a secured and decentralized machine learning approach applicable for large-scale data that can be trained locally [37]–[39]. Training the data locally might create a biased machine learning model as the training data can be specific for a particular region. For a machine learning model to be generalized, training data should be independent and identically distributed (IID), i.e., the probability of occurrence of each of the mutually independent random variables are same. However, the data at geographically dispersed locations could be non-IID [40], creating a biased model that fits the specific non-IID data. Inertia estimation when performed in a decentralized fashion involves non-IID data at several geographically dispersed locations. An FL approach can deal with both IID and non-IID data effectively [41], and hence the task of inertia estimation is explored for both cases in this article. The main contributions of this article are as follows.

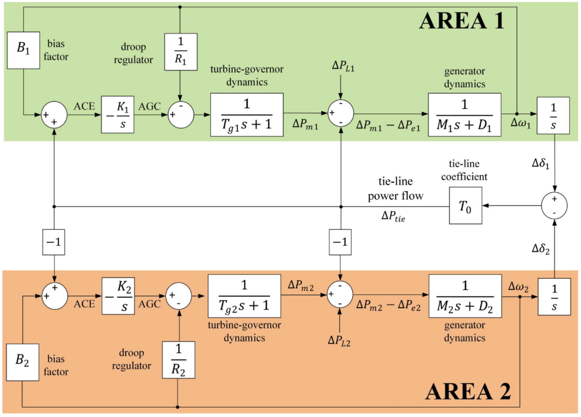


Fig. 1. Transfer function-based model of a two-area system.

1) Designed a CNN-based inertia estimator, in which the inertia constant of a particular area is estimated using local frequency measurements obtained from the PLL of an ESS. Because the estimation method is independent of the generating unit, it could also account for other grid assets that provide FFR to the grid but are ignored, such as synchronous motors or small DER equipped with virtual inertia.
2) Proposed a client–server-based framework, in which multiarea inertia is estimated in a decentralized fashion. The estimation task is performed locally and is an offline process, whereas the CNN weights are updated using the *federated averaging* [41] algorithm in an online fashion. To represent IID data, the entire dataset is randomly shuffled and redistributed to each of the ESS clients. To represent non-IID data, the training data was arranged in a particular order and specific parts of those orders were partitioned for the ESS clients. The proposed estimation framework is tested on both IID and non-IID data.
3) Improved the performance of the estimation framework by highlighting the trade-offs between the number of communication rounds and number of local epochs.

C. Organization

The rest of this article is structured as follows. Section II describes the frequency response in a multiarea power system. In Section III, FL-based multiarea inertia estimation technique is presented. Section IV describes the simulation setup, and the results and analysis are presented in Section V. Section VI concludes this article.

II. Frequency Response of Multiarea Power Systems

This section provides a general description of multiarea power system and its frequency response. For experimental purposes, we have considered a two-area power system, and it is assumed that the transfer function-based model represents the dynamics of an actual two-area power system model. Furthermore, we also assume that the overall generating units in an area can be represented by an equivalent generator. This section also discusses system perturbation using excitation signals from an ESS.

A. Multiarea Power Systems

Fig. 1 shows a transfer function-based model of Kundur's two-area system [16], [42].[1] It is assumed that the simplified transfer function-based model represents the overall dynamics of the two-area power system. As opposed to the single area equivalent generator system, a two-area power system is interconnected via tie-lines that are connected with the secondary control of each of the areas. The tie-line power is obtained as the difference between the power angle ($\Delta\delta$) of the generating units. The inertial response is the result of change in frequency ($\Delta\omega$), which is ultimately related to the difference in mechanical to electrical power ($\Delta P_m - \Delta P_e$). Hence, in a multiarea system, the effect of tie-line power can affect the inertial response of individual areas as the tie-line power compensates the change in rotor angle of the generating units. The effect of tie-line is given by the area control error (ACE), which is the difference between $\Delta\omega$ and the bias factor (B). ACE also takes into account the error between scheduled and actual power transfer between areas. Furthermore, K is the integral gain of the automatic generation control, T_g is the turbine-governor time constant, ΔP_L is the change in load, M is the inertia constant, D is the damping coefficient, R is the speed regulation droop, and T_0 is the tie-line coefficient. The mathematical relationship and in-depth explanation of each of these parameters are given in [16] and [42].

B. Inertia Estimation Using an Excitation Signal

Power system inertia estimation can be conducted using excitation signals that are generated through the power electronic interface of ESS without affecting system stability [6], [18]. Utility-scale ESSs are expected to be widely deployed in power systems for various services [43]. Hence, such existing power electronic-based ESS can be used to perturb the system and, based on the local frequency measurements obtained from the PLL of the ESS, inertia estimation can be performed [44], [45]. In this article, pulsating excitation signals of fixed frequency and varying magnitudes are used. However, it should be noted that to collect the snapshots, only the area in which the snapshots are to be collected is perturbed, while the change in load for the other area is assumed to be zero, i.e., the data collection procedure is asynchronous. A sample excitation signal, fed to Area 1 of the system in Fig. 1 with an amplitude of ΔP_{L1}, and corresponding measurements of $\Delta\omega_1$ and $\Delta\dot{\omega}_1$ are shown in Fig. 2.

The sampling time and sampling frame of the measurements can be defined after collecting the snapshots from the PLL of the local ESS. The noise in $\Delta\omega_1$ and $\Delta\dot{\omega}_1$ represents normal Gaussian measurement noise [46]. Because we are interested in estimating the inertia constant, we only consider the sampling frame, in which the inertial response is prominent. By varying M and ΔP_L for individual areas, multiple snapshots can be collected for training. To make it practical, the frequency measurements can be collected when there is a new dispatch in the power system. At that point, the measured data can be used for the training purpose locally, and the network gradients are updated whenever the clients and the server communicates. The procedure is discussed in detail in the remaining sections. For the purpose of this research, it is assumed that M and ΔP_L are varied in a simulated environment.

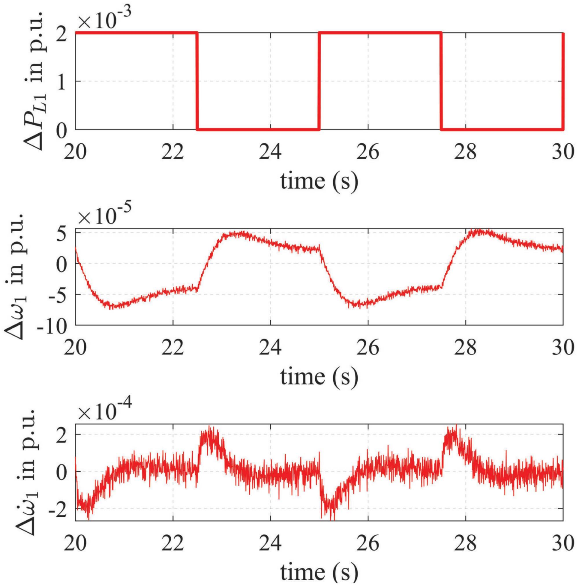


Fig. 2. Sample of excitation signal, frequency, and ROCOF measurements at Area 1 when $M_1 = 9$ s, $M_2 = 5$ s, $\Delta P_{L1} = 2 \times 10^{-3}$ p.u., and $\Delta P_{L2} = 0$ p.u. Only the area in which the frequency snapshots are to be collected is perturbed via excitation signal. In this case, Area 1 is perturbed via $\Delta P_{L1} = 2 \times 10^{-3}$ p.u., whereas $\Delta P_{L2} = 0$ p.u.

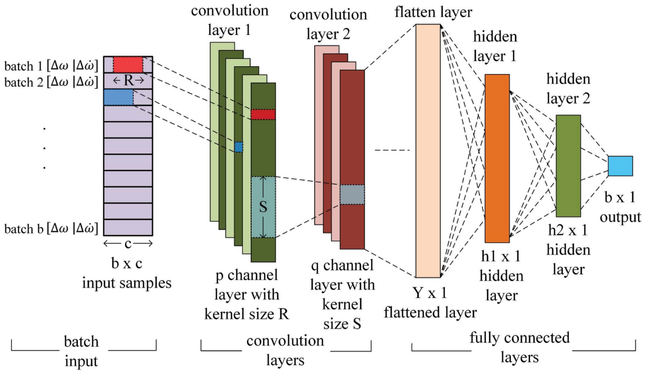


Fig. 3. Inertia estimation using 1-D CNN trained in batches.

III. Multiarea Inertia Estimation Using FL

This section describes an FL-based decentralized approach to estimate inertia using local frequency measurements. A CNN-based inertia estimation method is described that takes local noisy frequency measurements as its input and estimates the inertia. It is assumed that the system has a preexisting communication protocol between the clients and the server. The design and description of such protocol is not within the scope of this work.

A. CNN-Based Inertia Estimation

Fig. 3 shows a 1-D CNN that takes the frequency measurements, $\Delta\omega$ and $\Delta\dot{\omega}$, as inputs and estimates the inertia constant.

[1] Subscript 1 denotes Area 1 and subscript 2 denotes Area 2, unless stated otherwise, throughout the rest of this article.

The measurements $\Delta\omega$ and $\Delta\dot{\omega}$ are stacked horizontally to create a single input vector of size c. Furthermore, the input is taken in the form of randomly selected batches of size b as CNNs are expected to perform well when trained in batches [47]. For each epoch, the entire batch of training samples are fed to the CNN, and the process is repeated for several epochs until a desired model accuracy is obtained. For each training iteration, the input is of the size $b \times c$ and the output will be a vector of size b with inertia estimates for the corresponding samples of $\Delta\omega$ and $\Delta\dot{\omega}$. One needs to define several hyperparameters (parameters with values that are set before the training process) for CNN that can be tuned to improve the performance of the CNN. CNN contains the kernels (i.e., filters) that slide through the input samples and get the activation for the respective convolution layer. The convolution layers in CNN exploit spatial patterns in the input data that can drastically improve the performance of the estimation method [48]. This feature can be beneficial to identify the region where the effect of system inertia is prominent in the frequency snapshots. The kernels are selected to be vectors of sizes $\mathcal{R}$ and $\mathcal{S}$, respectively, for two convolution layers. Let p and q represent the number of channels in each of the convolution layers. Because the input is of a single dimension, the number of input channels is one. At the end of the convolution layers begins a feedforward neural network that is trained to minimize the mean squared error (MSE) between the actual and estimated values using backpropagation [49]. In this work, two hidden layered feedforward neural networks are used with h_1 and h_2 hidden layers, respectively.

In this article, each of the ESS clients contain a CNN estimator that estimates the area inertia using local frequency measurements obtained from the PLL of the ESS. To assess the performance of the estimator, this article analyzes both the MSE and root MSE (RMSE) values. MSE computes the square of the errors, whereas RMSE is in the same scale as the error values. MSE and RMSE are chosen as error metrics over other metrics due to their ability to penalize larger errors by squaring the error, which ensures faster convergence of the CNN model.

B. FL-Based Inertia Estimation

Existing centralized machine learning approaches require training data that is collected in a centralized location and perform the prediction on a single model based on the aggregated training data [41]. This method is expensive and inefficient from a communication, memory, and cybersecurity point of view—the central server needs to have a large storage capacity to accommodate entire training data from the clients, and the clients have to continually communicate to update the training data. Hence, the information can be breached while communicating the data, which is not desirable considering the increasing number of cyber threats in the power system [50], [51].

FL is a secure and robust framework that facilitates decentralized machine learning. The training is performed remotely on individual clients in a decentralized fashion [52]. However, after the training has been completed, a central server aggregates the trained weights from each of the clients and then redistributes the aggregated weights to the clients. FL is highly efficient, and is more robust than conventional machine learning techniques due to following reasons.

1) Only model weights are communicated from the clients to server and vice-versa. This ensures data privacy and discourages possible cyberattacks, which is a serious concern in the field of power systems [50], [51].
2) The training process at each of the clients is offline, whereas the communication between the clients and server occurs only during the weight aggregation and distribution phase, if the clients are available. Communication cost is important in any online optimization task as the bandwidth of communication could be limited. In FL, the number of local training epochs can be varied to improve the model at client's level. This can reduce the communication cost drastically.
3) Because the weights of the neural network are just floating point values, the server does not require excess memory to store and aggregate the weights of the clients. This is an important aspect of FL that makes it more data-efficient compared to other machine learning algorithms.

FL can be applied in a multiarea system to estimate the inertia constant of individual areas. It should be noted that FL does not estimate the inertia, but rather provides a framework to do so in a decentralized fashion. In FL, the training data (i.e., frequency snapshots) reside at the client ESS location, making it a decentralized machine leaning algorithm. Each of the areas will have a shared CNN model that estimates the inertia constant at that particular area as described in the former section. Fig. 4 shows a general framework of FL-based inertia estimation in a multiarea power system. Each of the communication rounds between the server and the clients can be represented by three processes—check-in, configuration and training, and weight aggregation. Let N be the total number of ESS clients, each belong to an Area ξ. During check-in, a fraction C of N clients ($C \leq N$) are randomly selected by the server. The red cross denotes the clients that are not selected, or may be offline, during the check-in phase at a particular round. However, the nonselected clients in the former communication rounds are also selected in the subsequent communication rounds once they come back online. Let Ψ_i be the set of m ESS clients for round i. Here, $m = \max(C \times N, 1)$ is the number of selected clients, ensuring at least one client is selected. During the configuration and training process, the server initializes the weight of the shared CNN model, $w_0 \leftarrow w_t$, and distributes it to the selected clients in Ψ_i. Initializing a common weight in the server is found to be more effective than random initialization of weights in each of the clients [41], [53].

The learning rate (α) and local mini-batch size (b) are defined at each of the ESS clients n in particular area ξ. The number of local epochs, $\mathbb{E}$, can be varied to achieve the best performance. Furthermore, $\mathbb{E}$ can be varied for individual clients, i.e., CNN at different clients can be trained on local frequency measurements for a different number of epochs. Because the server does not keep track of number of training epochs at the CNN of each client n, it is reasonable to posit that in a real world scenario $\mathbb{E}$ will be different for different clients. However, for simplicity, in this work $\mathbb{E}$ is consistent for each of the clients. Let $\mathbb{P}_n$ be the set

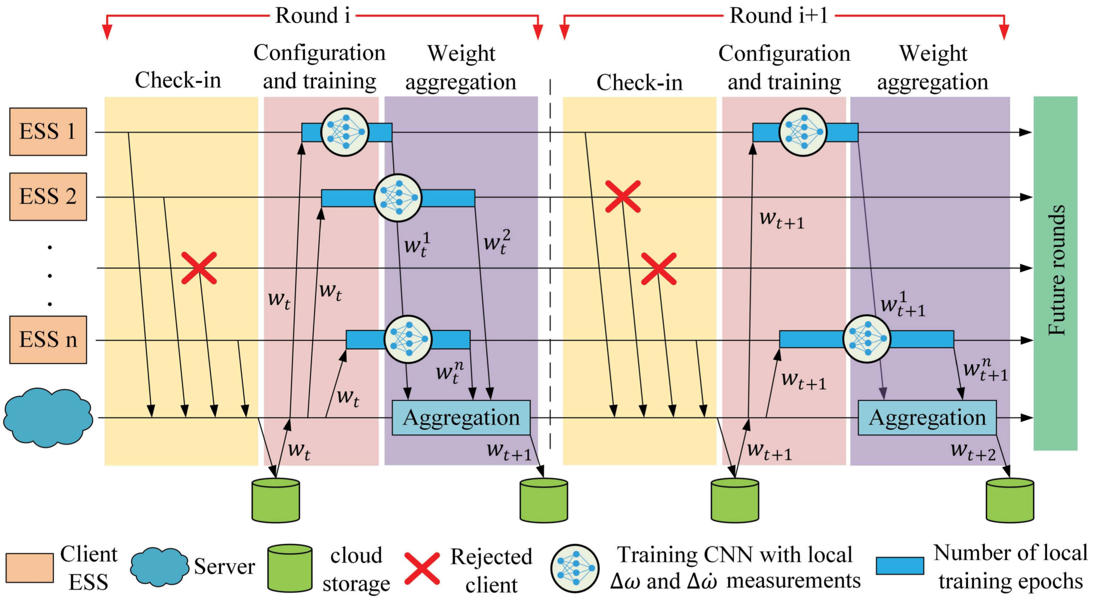


Fig. 4. FL-based framework for inertia estimation in a multiarea power system. The cloud storage is some form of storage device that stores the aggregated CNN model and information about selected clients.

of training snapshots at ESS client n, $\mathbb{B}$ represents the batches of training snapshots where each batch $\beta \in \mathbb{B}$ is of size b. At each ESS clients n, the CNN model then estimates the inertia constant by minimizing MSE via backpropagation.

Finally, when the clients are online, the trained weights of each of the clients, w_{t+1}^n, are sent back to the server for aggregation. In this work, the weight aggregation is performed once the server receives the trained weights of all selected clients. Weight aggregation is the most important process in a communication round that makes FL different than other machine learning techniques. After collecting the trained weights, w_{t+1}^n, from all clients in Ψ_i, a weighted average method based on the number of data samples on each of the clients is given by

$$w_{t+1} = \sum_{n=1}^{N} \frac{\mu_n}{\mu} w_{t+1}^n \qquad (1)$$

where μ_n is the number of training snapshots at client n and μ is the total number of training snapshots for N clients. The server stores the aggregated weight, w_{t+1}, in its persistent storage as a checkpoint and the entire process is repeated for the next round, $i+1$. The algorithm for FL is known as federated averaging due to its unique weight averaging method given in (1). The pseudocode for federated averaging is given in Algorithm 1.

C. Overall Framework for Multiarea Inertia Estimation

A schematic of the overall framework of inertia estimation in multiarea system using FL is shown in Fig. 5. A two-area power system model connected by a tie-line is used in this work. The power system at each area is asynchronously perturbed by excitation signal ΔP_L. The perturbation is fed only at one of the areas that the frequency snapshots are to be observed, while keeping the perturbation at the other area to zero, i.e., in Fig. 5 ΔP_{L1} is an excitation signal with a given amplitude and frequency whereas $\Delta P_{L2} = 0$. The additional Gaussian noise signal is added in the measurement to mimic noisy PLL measurements. The CNN located at each of ESS clients improves the estimation by minimizing MSE between actual value (M) and estimated value ($\tilde{M}$), and updates the model parameters via backpropagation. However, the local weight updates corresponding to each of the clients, represented by w_t^1 and w_t^2, are different. The trained updates are sent to the server for aggregation via a secured communication channel. The server then aggregates the weight and distributes the shared model to each of the ESS clients. This process repeats for several communication rounds until the global model converges.

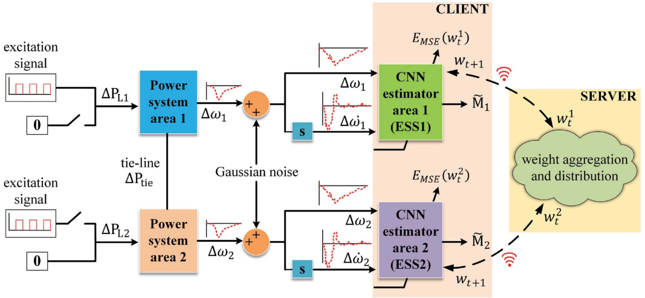


Fig. 5. Overall schematic of FL-based inertia estimation in a multiarea power system. Note that only one of the areas is perturbed at a time via ΔP_L to collect the snapshots.

IV. Simulation Setup

A. Overview

The modeling and simulation of the multiarea power system, along with data collection and preprocessing, was conducted in MATLAB/Simulink 2018b. The CNN model and FL framework

Algorithm 1: Federated Averaging for Inertia Estimation.

```
1  initialize the number of ESS clients $N$, fraction of
     ESS clients to be selected $C$, number of local epochs
     at each of the ESS clients $\mathbb{E}$, local mini-batch size $b$,
     and learning rate $\alpha$
2  Server:
3      initialize the weight of CNN $w_0 \leftarrow w$
4      for each round $i = 1, 2....$ do
5          max ESS clients selected $(m) \leftarrow$ max($C \times N$,1)
6          $\Psi_i \leftarrow$ random set of $m$
             ESS clients for each round
7          for each ESS client $n \in \Psi_i$ do
8              // call function at line 14
9              gradient $(w_{t+1}^n) \leftarrow$ ESS_Client($n, w$)
10         end
11         aggregate weight, $w_{t+1} \leftarrow \sum_{n=1}^{N} \frac{\mu_n}{\mu} w_{t+1}^n$
12         distribute aggregated weight to the clients in $\Psi_i$
13     end
14 Function ESS_Client($n, w$):
15     // this function will run on each of the ESS Clients
         $n$
16     $\mathbb{B} \leftarrow$ splitting entire training snapshots, $\mathbb{P}_n$, into
         batches of size $b$
17     for 1 to $\mathbb{E}$ do
18         for batch $\beta \in \mathbb{B}$ do
19             // perform back propagation
20             $w_{t+1}^n \leftarrow w_t^n - \alpha \nabla E_{MSE}$
21         end
22     end
23     return $w_{t+1}^n$
```

was developed in Python using PyTorch, an open-source library for deep learning studies [54]. To leverage the fast computing abilities of PyTorch, the machine learning model was trained on South Dakota State University's Roaring Thunder cluster on NVIDIA Tesla P100/V100 GPUs. Although GPUs were used to train this model for speed of analysis, modern microcontrollers with ARM cortex cores have been successful in training deep CNN architectures and can be used in real-world implementations [55].

B. Simulation Benchmark

The transfer function-based two-area system with an equivalent generator model, shown in Fig. 1, was used as an experimental model to collect the frequency snapshots, and the respective simulation parameters are given in Table I [16], [42], [56]. To have variation in the dataset, the snapshots were collected from both the areas with different values of M_1 and M_2. Similarly, for each of the areas, excitation signals with 100 different values of ΔP_L from 10^{-3} to 0.1 p.u. with an increment 10^{-3} p.u. were used. As mentioned before, the excitation signals were fed only in the area the snapshots were collected, as shown in Fig 5.

To collect realistic data samples, white Gaussian noise was introduced in the signal using add white Gaussian noise (AWGN) block in MATLAB/Simulink. The signal-to-noise ratio of 45 dB with a covariance of 10^{-6} was found to be appropriate for our setup as described in [46]. A total of 900 snapshots were collected from both of the areas with a sampling frequency of 200 Hz, as mentioned in the IEEE standard [57]. Similarly, a sampling frame (inertial response time frame) of 1 s was used—from 31 to 32 s—as the system took time to reach a steady–steady response to the excitation signal.

TABLE I
Simulation Parameters for Multiarea Power System

Parameters	Values
inertia constant for area 1 (M_1)	$[2, 4, 6, 8]$ s
inertia constant for area 1 (M_2)	$[3, 5, 7, 9, 10]$ s
change in load ($\Delta P_{L1}, \Delta P_{L2}$)	10^{-3} p.u. – 0.1 p.u. with a step of 10^{-3} p.u.
damping coefficient (D_1, D_2)	1.5
speed regulation droop for area 1 (R_1)	5%
speed regulation droop for area 2 (R_2)	10%
turbine-governor time constant (T_{g1}, T_{g2})	0.2 s
secondary controller gain (K_1, K_2)	$4.65e - 2$
bias factor for area 1 (B_1)	21.5
bias factor for area 2 (B_2)	11.5
tie-line coefficient (T_0)	1

C. Data Distribution and Hyperparameters Selection for FL

A sampling time of 200 Hz gives 200 data points ($c = 400$) for each of the snapshots of $\Delta\omega$ and $\Delta\dot{\omega}$, extracted at a sampling frame of 1 s. To have a comparison on the basis of communication rounds and architecture, the simulation was performed on both CNN and a CNN architecture, multilayer perceptron (MLP), for which the architecture and hyperparameters were selected, as described in [18]. This section describes a general method of data partitioning for IID and non-IID cases and hyperparameter selection for the FL framework.

To test the global model on a validation set for each communication round, the overall data was split into two parts — 720 snapshots ($\sim 80\%$) for training and 180 snapshots ($\sim 20\%$) for validation. The general method of having an IID case would be to distribute the snapshots to each of the areas so as to have a near equal probability distribution for each value of inertia constant. To achieve this, the training dataset was randomly shuffled and redistributed so that each ESS client contained 350 random snapshots. For the non-IID case, the 720 training snapshots were arranged in ascending order of M and distributed in equal parts to the ESS clients. This ensure that the two ESS clients have snapshots corresponding to different values of M, e.g., the snapshots corresponding to $M = 2$ s is in Area 1 but not in Area 2.

The effectiveness of the federated averaging algorithm depends on three hyperparameters — C, $\mathbb{E}$, and b [41]. Because $N = 2$, we selected a value of $C = 1$ that means that all of the clients are selected during each communication round. Furthermore, we experimentally verified that $b = 10$ works well for both CNN and MLP. When some level of accuracy is desired from the global model, the algorithm can be stopped at the particular communication round that the best global accuracy is obtained. However, for inertia estimates, we assume that an estimated value within 10% of the actual value is a correct value.

TABLE II
RMSE OF ESTIMATED VALUES OF INERTIA ON VALIDATION SET FOR IID DATA

MLP			CNN		
$i = 1000$ $\mathbb{E} = 1$	$i = 200$ $\mathbb{E} = 5$	$i = 100$ $\mathbb{E} = 10$	$i = 200$ $\mathbb{E} = 1$	$i = 100$ $\mathbb{E} = 2$	$i = 40$ $\mathbb{E} = 5$
0.4387	**0.4482**	**0.4484**	**0.3652**	**0.3534**	**0.3914**

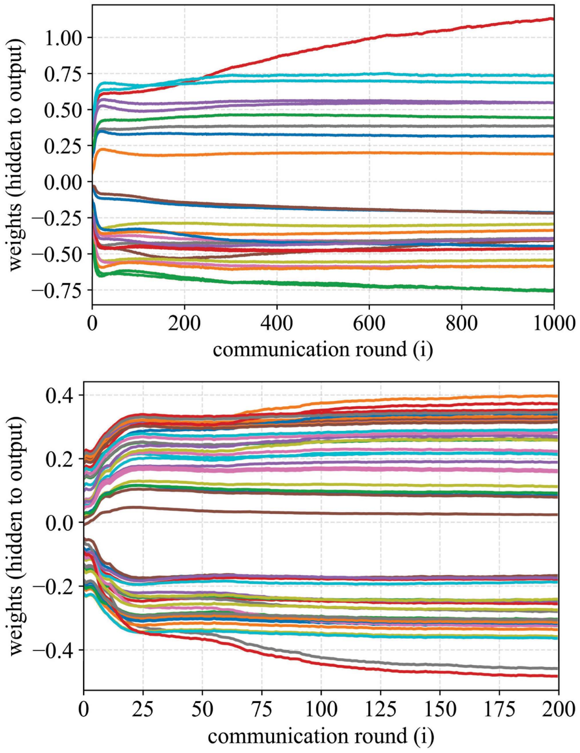


Fig. 6. Evolution of aggregated weight, w_{t+1}, using federated averaging for each communication round i between the server and ESS clients for MLP-based training (top) and CNN-based training (bottom). The plots represent the IID case when $\mathbb{E} = 1$. MLP (top) takes $\sim$ 1000 iterations to reach the desired accuracy.

Hence, we predefined i before conducting the simulation in this work. The simulation conducted for different values of $\mathbb{E}$ and i for MLP and CNN are presented in Section V. The combinations of i and $\mathbb{E}$ are so chosen to get a similar value of RMSE on the validation set.

V. RESULTS AND ANALYSIS

A. Performance Metrics for IID Data

For the IID case, the RMSE values on the validation set of 180 snapshots for MLP and CNN are given in Table II. The presented value is the RMSE observed for the ith communication round, where the neural networks at the ESS clients are trained for $\mathbb{E}$ number of local epochs on IID data. When $\mathbb{E} = 1$, it takes 200 communication rounds between the server and the ESS clients to achieve an RMSE of 0.3652 when trained using a CNN. However, when trained using the MLP, an RMSE of 0.4387 is obtained with 1000 communication rounds. To get a similar level of model performance, MLP requires a much higher number of client–server communication than with CNN.

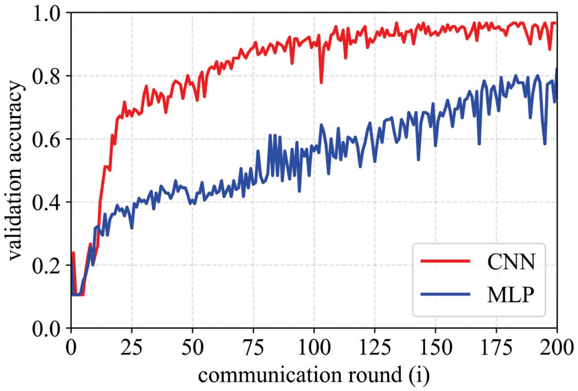


Fig. 7. Aggregated model accuracy on validation set for each communication round i between the server and ESS clients for MLP-based and CNN-based training. The plot represents the IID case when $\mathbb{E} = 1$. MLP takes $\sim$ 1000 communication rounds to reach a desired level of accuracy as compared to $\sim$ 200 communication rounds for CNN.

TABLE III
RMSE OF ESTIMATED VALUES OF INERTIA ON VALIDATION SET FOR NON-IID DATA

MLP			CNN		
$i = 1000$ $\mathbb{E} = 1$	$i = 200$ $\mathbb{E} = 5$	$i = 100$ $\mathbb{E} = 10$	$i = 200$ $\mathbb{E} = 1$	$i = 100$ $\mathbb{E} = 2$	$i = 40$ $\mathbb{E} = 5$
0.3851	**0.372**	**0.3876**	**0.3982**	**0.5372**	**0.401**

Fig. 6 shows the evolution of aggregated weights of the global model, obtained via federated averaging, for IID case when $\mathbb{E} = 1$. It can be seen that the weights are saturated at $\sim i = 175$ for CNN (approximately 5.7 times less than MLP), whereas some of the weights do not converge for MLP-based model even when $i = 1000$.

Similarly, Fig. 7 shows the accuracy of the aggregated model on validation set at the end of each communication rounds for the IID case when $\mathbb{E} = 1$. CNN-based approach reaches the desired accuracy in less number of communication rounds than the MLP-based approach. The MLP-based model gave a validation accuracy of 95% at $i = 1000$, and CNN gave a validation accuracy of 96.67% at 200 epochs. Hence, based on the abovementioned results, the CNN-based estimator outperforms the MLP in terms of communication cost and RMSE for IID data.

B. Performance Metrics for non-IID Data

For the non-IID case, the RMSE values on the validation set of 180 snapshots for MLP and CNN are given in Table III. The presented value is the RMSE observed for ith communication round, where the neural networks at the ESS clients are trained for $\mathbb{E}$ number of local epochs on non-IID data. When $\mathbb{E} = 1$, it takes 200 communication rounds between the server and the ESS clients to achieve an RMSE of 0.372 when trained using CNN. However, using MLP, an RMSE of 0.3851 is obtained after 1000 communication rounds.

Fig. 8 shows the evolution of aggregated weights of the global model, obtained via federated averaging, for non-IID case when $\mathbb{E} = 1$. Similar to the IID case, it can be seen that the weights are

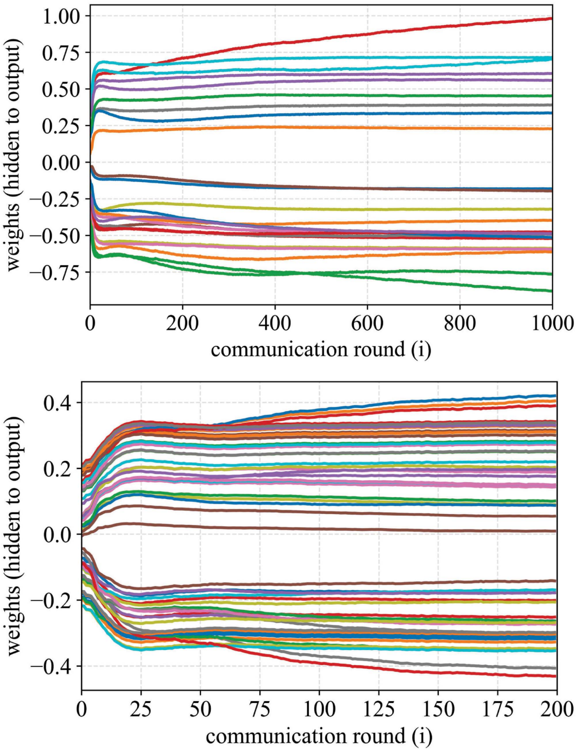


Fig. 8. Evolution of aggregated weight, w_{t+1}, using federated averaging for each communication round i between the server and ESS clients for MLP-based training (top) and CNN-based training (bottom). The plots represent the non-IID case when $\mathbb{E} = 1$. MLP (top) takes $\sim$ 1000 iterations to reach the desired accuracy.

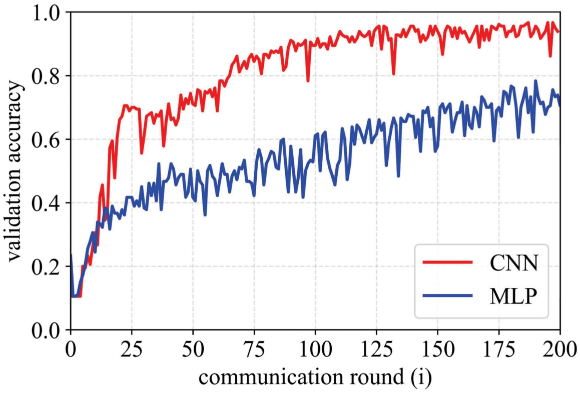


Fig. 9. Aggregated model accuracy on validation set for each communication round i between the server and ESS clients for MLP-based and CNN-based training. The plots represent the non-IID case for $\mathbb{E} = 1$. MLP takes $\sim$ 1000 communication rounds to reach a desired level of accuracy as compared to $\sim$ 200 communication rounds for CNN.

saturated at $\sim i = 175$ for CNN, whereas some of the weights do not converge for MLP-based model even when $i = 1000$. Similarly, Fig. 9 shows the accuracy of the aggregated model on validation set at the end of each communication rounds. Consistent with the results observed for the IID case, it takes less number of communication rounds to reach a desired accuracy for CNN-based approach as compared with the MLP-based approach.

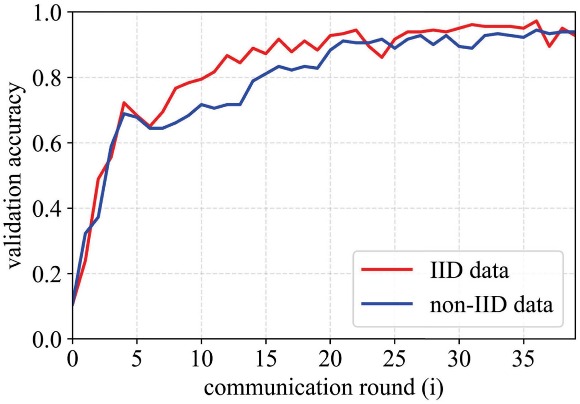


Fig. 10. Comparison of accuracy for IID and non-IID data when $i = 40$ and $\mathbb{E} = 5$.

From the above-mentioned analysis, it is interesting to observe that for two-area ESS clients, the performance metrics do not have much difference for the IID and the non-IID case. Fig. 10 shows the comparison of validation accuracy for IID and non-IID case when $\mathbb{E} = 5$. The convergence for IID case is slightly better than the non-IID case. This is due to the fact that IID data samples contain more training labels that improves the estimation of the overall model. However, with limited training labels for non-IID data, the performance of the estimation model is weaker compared with the model trained with IID data samples. It is also important to note that in this work we have distributed the data to the ESS clients by manually separating the data to IID and non-IID fashion in a controlled environment. Such scenario is not possible in real-world scenarios, and hence the results might not be generalizable to hundreds of ESS clients with highly non-IID data. The key takeaway from these analyses is that the global model is successful in estimating the multiarea inertia constant without being trained on several snapshots.

C. Communication Cost

Communication overhead incurs the highest optimization cost in FL [41]. Although FL rejects the clients that are unable to provide an update, or are offline during a particular instant of communication, the cost of communication overhead still overpowers the individual computational cost on the clients, as well as the cost to add an additional client in the framework [41], [52]. In FL, the communication cost can be drastically reduced by increasing the number of $\mathbb{E}$ to a certain extent. Fig. 11 shows the validation accuracy of a CNN model for IID data with respect to i for different values of $\mathbb{E}$. It can be seen that when the value of $\mathbb{E}$ is increased from 1 to 5, the number of communication rounds to achieve the desired accuracy reduces drastically. When $\mathbb{E} = 5$, only 40 communication rounds would suffice to achieve an accuracy beyond 95%. This is a decrease in communication round by a multiple of 5 as compared to the case when $\mathbb{E} = 1$. Furthermore, the CNN model with $\mathbb{E} = 1$ was approximately six times more computationally efficient than the MLP counterpart. Therefore, when $E = 5$, CNN-based FL framework is 30 times more efficient than the MLP-based framework. Hence, the FL

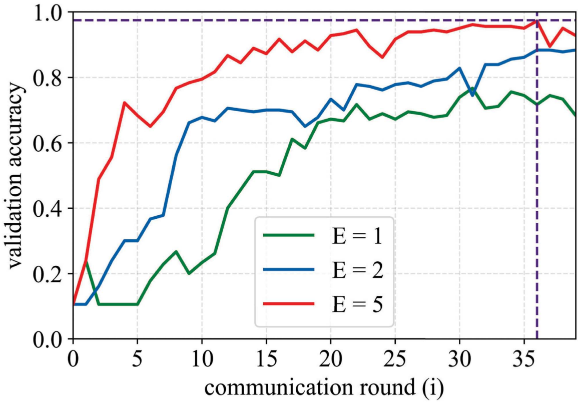


Fig. 11. Validation accuracy of a CNN-based model on IID data with respect to i for different values of $\mathbb{E}$. On varying the values of $\mathbb{E}$, the number of communication rounds can be controlled in FL.

learning framework can be made more efficient by decreasing the number of communication rounds between the server and the ESS clients and simultaneously increasing the value of $\mathbb{E}$.

Additionally, an early stopping technique can be used when the desired accuracy is obtained. In Fig. 11, the training at the client ESS and the communication between the client ESS and the server can be stopped when $i = 36$ (represented by a dashed vertical red line with a horizontal intersecting line showing the equivalent accuracy) to get the maximum accuracy.

D. Comparison with Existing Methods

The approach detailed in [32] presents a disturbance-based inertia estimation method while accommodating the dynamics of voltage and frequency of the power system. The authors model the effect of a power disturbance and frequency measurements to estimate the inertia of the system. However, the estimation task is centralized as the disturbance and dynamics data are centrally collected to estimate the inertia. Furthermore, the method is typically presented for synchronous generator models and may not perform well for nonsynchronous units. The methods in [34] and [35] involve data-driven approaches to estimate the system inertia. In [34], a neural network-based inertia forecast tool is presented in a system with high penetration of wind farms. However, the method is centralized and has no provision for non-IID data. In addition, further analysis is required to identify the variables correlated with system inertia to improve the performance of the neural network-model. Similarly, Yang *et al.* [35] presented a modal identification-based inertia estimation approach. Similarly, Cai *et al.* [27] performed eigenvalue analysis on the power oscillation and frequency signals obtained from the PMU. As mentioned before, such electromechanical modes of oscillation change drastically when nonsynchronous units are connected in the system. Hence, the proposed method is only applicable for systems dominated by synchronous generating units.

The existing methods discussed above lack several features that highlight the benefit of the proposed method. The essential feature of sharing CNN weights among the clients, rather than the training data, makes the proposed approach secure. Furthermore, none of the existing methods handle non-IID data for inertia estimation. The overall comparison of the proposed approach against existing methods in [32], [34], and [35] is given in Table IV.

TABLE IV
COMPARISON OF PROPOSED APPROACH WITH [32], [34], AND [35]

Attributes	Proposed aproach	Swing equation model [32]	neural network [34]	Modal identification [35]
decentralized estimation	✓	×	×	×
Generating unit independence	✓	×	✓	×
Consistency with input noise	✓	✓	✓	✓
Applicability with non-IID data	✓	×	×	×
Provision for cybersecurity	✓	×	×	×

VI. Conclusion

In this article, the inertia constant was estimated in a multi-area power system using a federated averaging algorithm. The simulation was conducted and verified for two neural network architectures—MLP and CNN. The frequency snapshots were collected at the PLL of each ESS client using nonintrusive excitation signals. It was found that MLP takes a greater number of communication rounds compared to CNN to get a similar level of accuracy. Furthermore, the framework was verified to perform well for both IID and non-IID data with significant accuracy, which is important in the field of power systems that contains highly non-IID data. Both the MLP and CNN-based inertia estimators showed good performance even with noisy input samples. It was also verified that the number of communication rounds between the ESS clients and the server can be drastically reduced by increasing the number of local epochs, $\mathbb{E}$ at each of the clients. The number of communication round reduced drastically when using a CNN model with higher number of local epochs as compared to the MLP model. Traditional data-driven estimation methods are increasingly unsuitable for modern power grids inundated with Big Data of non-IID nature. Hence, the ability of FL can be leveraged to perform estimation tasks in such environment without collecting the data from decentralized locations. This tool can be highly useful in areas that traditionally involve data collection from multiple areas, such as energy demand prediction. A potential direction for future research is to estimate the unknown inertia in the system considering its time-varying characteristics. The current work focused mostly on the decentralized method for data-driven inertia estimation, without considering any time-varying nature of inertia. However, the problem can also be formulated as a time-series problem and can be integrated with the proposed decentralized strategy.

Acknowledgment

The authors would like to thank Dr. Imre Gyuk, director of the Energy Storage Program, for his continued support.

REFERENCES

[1] F. Milano, F. Dörfler, G. Hug, D. J. Hill, and G. Verbič, "Foundations and challenges of low-inertia systems (*invited paper*)," in *Proc. Power Syst. Comput. Conf.*, 2018, pp. 1–25.

[2] M. Nedd, J. Browell, K. Bell, and C. Booth, "Containing a credible loss to within frequency stability limits in a low-inertia GB power system," *IEEE Trans. Ind. Appl.*, vol. 56, no. 2, pp. 1031–1039, Mar./Apr. 2020.

[3] H. Sturgess, W. Mak, and A. Bourn, "Initial analysis and thoughts on the power outages on 9th August," Aurora Energy Res., Oxford, U.K., Tech. Rep., Aug. 2019.

[4] AEMO, "Black system South Australia 28 Sep. 2016," Australian Energy Market Operator, Melbourne, VIC, Australia, Tech. Rep., Mar. 2017.

[5] Pakistan power cut plunges country into darkness, BBC News, Jan. 2021. [Online]. Available: https://www.bbc.com/news/world-asia-55606598

[6] J. Zhang and H. Xu, "Online identification of power system equivalent inertia constant," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 8098–8107, Oct. 2017.

[7] F. Allella, E. Chiodo, G. M. Giannuzzi, D. Lauria, and F. Mottola, "On-line estimation assessment of power systems inertia with high penetration of renewable generation," *IEEE Access*, vol. 8, pp. 62689–62697, 2020.

[8] P. Makolo, R. Zamora, and T.-T. Lie, "Online inertia estimation for power systems with high penetration of RES using recursive parameters estimation," *IET Renewable Power Gener.*, vol. 15, pp. 2571–2585, Sep. 2021.

[9] A. Poudyal, "Distributed machine learning approach to fast frequency response-based inertia estimation in low inertia grids," M.S. thesis, Elect. Eng. Comput. Sci., South Dakota State Univ., Brookings, SD, USA, Aug. 2020.

[10] National Renewable Energy Laboratory, "Inertia and the power grid: A guide without the spin," USA: Golden, CO., Tech. Rep. NREL/TP-6A20-73856, May 2020. [Online]. Available: https://www.nrel.gov/docs/fy20osti/73856.pdf

[11] Y. Su *et al.*, "An adaptive PV frequency control strategy based on real-time inertia estimation," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2355–2364, May 2021.

[12] ERCOT, "Inertia: Basic concepts and impacts on the ERCOT grid," Electric Rel. Council Texas, Austin, TX, USA, Tech. Rep., Apr. 2018.

[13] U. Tamrakar *et al.*, "Virtual inertia: Current trends and future directions," *Appl. Sci.*, vol. 7, no. 7, Jun. 2017, Art. no. 654.

[14] H. Golpîra, A. R. Messina, and H. Bevrani, "Emulation of virtual inertia to accommodate higher penetration levels of distributed generation in power grids," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3384–3394, Sep. 2019.

[15] U. Tamrakar, T. M. Hansen, R. Tonkoski, and D. A. Copp, "Model predictive frequency control of low inertia microgrids," in *Proc. IEEE 28th Int. Symp. Ind. Electron.*, 2019, pp. 2111–2116.

[16] P. Kundur, N. J. Balu, and M. G. Lauby, *Power System Stability and Control*, vol. 7. New York, NY, USA: McGraw-Hill, 1994.

[17] S. Rehimi, R. Mirzaei, and H. Bevrani, "Interconnected microgrids frequency response model: An inertia-based approach," *Energy Rep.*, vol. 6, pp. 179–186, Feb. 2020, doi: 10.1016/j.egyr.2019.11.060.

[18] A. Poudyal, R. Fourney, R. Tonkoski, T. M. Hansen, U. Tamrakar, and R. D. Trevizan, "Convolutional neural network-based inertia estimation using local frequency measurements," in *Proc. 52nd North Amer. Power Symp.*, 2021, pp. 1–6.

[19] L. Lugnani *et al.*, "ARMAX-based method for inertial constant estimation of generation units using synchrophasors," *Electric Power Syst. Res.*, vol. 180, Mar. 2020, Art. no. 106097.

[20] R. K. Panda, A. Mohapatra, and S. C. Srivastava, "Online estimation of system inertia in a power network utilizing synchrophasor measurements," *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 3122–3132, Jul. 2020.

[21] M. A. M. Ariff, B. C. Pal, and A. K. Singh, "Estimating dynamic model parameters for adaptive protection and control in power system," *IEEE Trans. Power Syst.*, vol. 30, no. 2, pp. 829–839, Mar. 2015.

[22] P. M. Ashton, C. S. Saunders, G. A. Taylor, A. M. Carter, and M. E. Bradley, "Inertia estimation of the GB power system using synchrophasor measurements," *IEEE Trans. Power Syst.*, vol. 30, no. 2, pp. 701–709, Mar. 2015.

[23] K. Tuttelberg, J. Kilter, D. Wilson, and K. Uhlen, "Estimation of power system inertia from ambient wide area measurements," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 7249–7257, Nov. 2018.

[24] J. Matevosyan, "Evolution of ERCOT's frequency control and ancillary services for higher levels of inverter based generation," Feb. 2019. [Online]. Available: https://www.esig.energy/resources/evolution-of-ercots-frequency-control-and-ancillary-services-for-higher-levels-of-inverter-based-generation-julia-matevosyan-february-2019/

[25] P. Du, N. V. Mago, W. Li, S. Sharma, Q. Hu, and T. Ding, "New ancillary service market for ERCOT," *IEEE Access*, vol. 8, pp. 178391–178401, 2020.

[26] X. Cao, B. Stephen, I. F. Abdulhadi, C. D. Booth, and G. M. Burt, "Switching Markov Gaussian models for dynamic power system inertia estimation," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3394–3403, Sep. 2016.

[27] G. Cai, B. Wang, D. Yang, Z. Sun, and L. Wang, "Inertia estimation based on observed electromechanical oscillation response for power systems," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4291–4299, Nov. 2019.

[28] M. Abdollahi, J. I. Candela, J. Rocabert, M. A. Elsaharty, and P. Rodrigue, "Novel analytical method for dynamic design of renewable SSG SPC unit to mitigate low-frequency electromechanical oscillations," *IEEE Trans. Power Electron.*, vol. 35, no. 7, pp. 7532–7544, Jul. 2020.

[29] S. You *et al.*, "Impact of high PV penetration on the inter-area oscillations in the U.S. Eastern interconnection," *IEEE Access*, vol. 5, pp. 4361–4369, 2017.

[30] A. Schmitt and B. Lee, "Steady-state inertia estimation using a neural network approach with modal information," in *Proc. IEEE Power Energy Soc. General Meeting*, 2017, pp. 1–5.

[31] D. Yang *et al.*, "Data-driven estimation of inertia for multiarea interconnected power systems using dynamic mode decomposition," *IEEE Trans. Ind. Inform.*, vol. 17, no. 4, pp. 2686–2695, Apr. 2021.

[32] D. Zografos, M. Ghandhari, and R. Eriksson, "Power system inertia estimation: Utilization of frequency and voltage response after a disturbance," *Electric Power Syst. Res.*, vol. 161, pp. 52–60, Aug. 2018.

[33] U. Tamrakar, D. A. Copp, T. A. Nguyen, T. M. Hansen, and R. Tonkoski, "Real-time estimation of microgrid inertia and damping constant," *IEEE Access*, vol. 9, pp. 114523–114534, 2021.

[34] E. S. N. R. Paidi, H. Marzooghi, J. Yu, and V. Terzija, "Development and validation of artificial neural network-based tools for forecasting of power system inertia with wind farms penetration," *IEEE Syst. J.*, vol. 14, no. 4, pp. 4978–4989, Dec. 2020.

[35] D. Yang *et al.*, "Ambient-data-driven modal-identification-based approach to estimate the inertia of an interconnected power system," *IEEE Access*, vol. 8, pp. 118799–118807, 2020.

[36] X. Zhang, F. Fang, and J. Wang, "Probabilistic solar irradiation forecasting based on variational Bayesian inference with secure federated learning," *IEEE Trans. Ind. Inform.*, vol. 17, no. 11, pp. 7849–7859, Nov. 2020.

[37] K. A. Bonawitz *et al.*, "Towards federated learning at scale: System design," in *Proc. Syst. Mach. Learn. Conf.*, 2019, pp. 1–15.

[38] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[39] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, "Energy demand prediction with federated learning for electric vehicle networks," in *Proc. IEEE Global Commun. Conf.*, 2019, pp. 1–6.

[40] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[41] H. B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[42] L. Fan, *Control and Dynamics in Power Systems and Microgrids*, vol. 1. Boca Raton, FL, USA: CRC Press, 2017.

[43] S. Stynski, W. Luo, A. Chub, L. G. Franquelo, M. Malinowski, and D. Vinnikov, "Utility-scale energy storage systems: Converters and control," *IEEE Ind. Electron. Mag.*, vol. 14, no. 4, pp. 32–52, Dec. 2020.

[44] U. Tamrakar, D. A. Copp, T. Nguyen, T. M. Hansen, and R. Tonkoski, "Optimization-based fast-frequency estimation and control of low-inertia microgrids," *IEEE Trans. Energy Convers.*, vol. 36, no. 2, pp. 1459–1468, Jun. 2021.

[45] M. Rauniyar *et al.*, "Evaluation of probing signals for implementing moving horizon inertia estimation in microgrids," in *Proc. IEEE North Amer. Power*, 2021, pp. 1–6.

[46] M. Brown, M. Biswal, S. Brahma, S. J. Ranade, and H. Cao, "Characterizing and quantifying noise in PMU data," in *Proc. IEEE Power Energy Soc. General Meeting*, 2016, pp. 1–5.

[47] E. Hoffer, I. Hubara, and D. Soudry, "Train longer, generalize better: Closing the generalization gap in large batch training of neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1731–1741.

[48] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol.*, 2017, pp. 1–6.

[49] M. Z. Alom *et al.*, “A state-of-the-art survey on deep learning theory and architectures,” *Electronics*, vol. 8, no. 3, Mar. 2019, Art. no. 292.

[50] C. Glenn, D. Sterbentz, and A. Wright, “Cyber threat and vulnerability analysis of the U. S. Electric Sector,” Idaho Nat. Lab., Idaho Falls, ID, USA, Tech. Rep. INL/EXT-16-40692, Aug. 2016.

[51] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, “Federated learning: A survey on enabling technologies, protocols, and applications,” *IEEE Access*, vol. 8, pp. 140699–140725, 2020.

[52] B. McMahan *et al.*, “Federated learning: Collaborative machine learning without centralized training data,” Apr. 2017. [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[53] I. J. Goodfellow, O. Vinyals, and A. M. Saxe, “Qualitatively characterizing neural network optimization problems,” in *Proc. Int. Conf. Learn. Representations*, 2015.

[54] A. Paszke *et al.*, “Automatic differentiation in PyTorch,” in *Proc. Neural Inf. Process. Syst.*, 2017.

[55] N. Suda and D. Loh, “Machine learning on arm Cortex-M microcontrollers,” Cambridge, U.K., ARM Ltd., White Paper, 2019. [Online]. Available: http://CRAN.R-project.org/package=raster

[56] G. Andersson, “Dynamics and control of electric power systems,” 2012. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.471.7315&rep=rep1&type=pdf

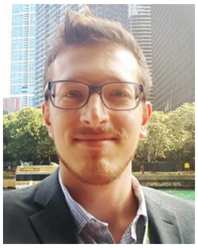
[57] *IEEE Standard for Synchrophasor Measurements for Power Systems*, IEEE Standard C37.118.1-2011 (Revision of IEEE Standard C37.118-2005), Dec. 2011.

Abodh Poudyal (Student Member, IEEE) received the B.E. degree from the Institute of Engineering, Tribhuvan University, Kirtipur, Nepal, in 2016 and the M.S. degree from South Dakota State University, Brookings, SD, USA, in 2020. He is currently working toward the Ph.D. degree with Washington State University, Pullman, WA, USA, all in electrical engineering.

His research interests include power distribution modeling and optimization, stochastic optimization, resilience assessment, modeling extreme events and their impact on the power grid, and AI-based applications on the power grid.

Ujjwol Tamrakar (Member, IEEE) received the B.E. degree from Tribhuvan University, Kirtipur, Nepal, in 2011 and the M.S. and Ph.D. degrees from South Dakota State University, Brookings, SD, USA, in 2015 and 2020, all in electrical engineering.

He is currently a Postdoctoral Appointee with Sandia National Laboratories, Albuquerque, NM, USA, and is also actively involved in various IEEE task forces related to voltage and frequency stability of power systems with high renewable penetration. His research interests include grid integration of renewable energy systems, energy storage analytics, power system modeling and stability, and optimal control.



Rodrigo D. Trevizan (Member, IEEE) received the B.Sc. degree in power systems engineering from the Grenoble Institute of Technology, Grenoble, France, in 2011, the B.S. and M.Sc. degrees in electrical engineering from the Federal University of Rio Grande do Sul, Porto Alegre, Brazil, in 2012 and 2014, respectively, and the Ph.D. degree in electrical engineering from the University of Florida, Gainesville, FL, USA, in 2018.

He has authored research papers on the subjects of control of energy storage systems and demand response for power grid stabilization, power system state estimation, and detection of nontechnical losses in distribution systems. His research interests include cyberphysical security, control, and valuation of energy storage systems.

Robert Fourney (Member, IEEE) received the B.S. degree in electrical engineering from Virginia Polytechnical Institute and State University, Blacksburg, VA, USA, in 1985, the M.S.E.E. degree in electrical engineering from the University of Maryland, College Park, College Park, MD, USA, in 1989, and the Ph.D. degree in engineering from the University of Maryland, College Park, College Park, MD, USA, in 2001.

He is currently an Associate Professor with Electrical Engineering and Computer Science Department, South Dakota State University, Brookings, SD, USA. His research interests include computer security, embedded systems, security for ubiquitous computing (IoT), sustainable power and energy systems, low-inertia power systems, and smart homes/smart cities.

Dr. Fourney has been the IEEE Siouxland Section Treasurer since 2019, and is the Faculty Advisor for the IEEE Student Section at South Dakota State University.

Reinaldo Tonkoski (Senior Member, IEEE) received the B.A.Sc. degree in control and automation engineering and the M.Sc. degree in electrical engineering from Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, Brazil, in 2004 and 2006, respectively, and the Ph.D. degree in electrical engineering from Concordia University, Montréal, QC, Canada, in 2011.

From 2009 to 2010, he was with CanmetENERGY, Natural Resources, Canada, where he worked on projects related to the grid integration of renewable energy sources. He is currently the Harold C. Hohbach Endowed Professor with the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD, USA, and a Visiting Faculty with Sandia National Laboratories, Albuquerque, NM, USA. He has authored more than 90 technical publications in peer reviewed journals and conferences. His research interests include grid integration of renewable energy systems and batteries, smart grid, and power electronics and control.

Dr. Tonkoski is the Editor of the IEEE TRANSACTIONS ON SUSTAINABLE ENERGY, and an Associate Editor for the IEEE ACCESS and IEEE SYSTEMS JOURNAL.

Timothy M. Hansen (Senior Member, IEEE) received the B.S. degree in computer engineering from the Milwaukee School of Engineering, Milwaukee, WI, USA, in 2011 and the Ph.D. degree in electrical engineering from Colorado State University, Fort Collins, CO, USA, in 2015.

He is currently an Associate Professor with the Electrical Engineering and Computer Science Department, South Dakota State University, Brookings, SD, USA. His research interests include optimization, high-performance computing, electricity market applications to sustainable power and energy systems, low-inertia power systems, smart cities, and cyberphysical-social systems.

Dr. Hansen is an active Member of ACM SIGHPC. He was the recipient of the 2019 IEEE-HKN C. Holmes MacDonald Outstanding Teaching Award and was the inaugural recipient of the Milwaukee School of Engineering Graduate of the Last Decade Award in 2020. He has been the IEEE Siouxland Section Chair since 2019, and is active within the IEEE PES Power Engineering Education Committee, where he is currently the awards Subcommittee Chair and Research Subcommittee Secretary-Elect.