A Federated Learning Approach for Distributed Solar Irradiance Forecasting

Cristiano G. M. Camandaroba*, Sophia S. G. Werneck*, Julia Z. C. de Mellos‡, Caian D. Jesus§, Matheus A. M. Ferreira¶, Frederico G. Guimarães‖, and Eduardo P. de Aguiar**
* *Department of Electrical Energy*, *Federal University of Juiz de Fora*, Juiz de Fora, Brazil
‡ *Graduate Program in Electrical Engineering*, *Federal University of Minas Gerais*, Belo Horizonte, Brazil
§ *Graduate Program in Computational Modeling*, *Federal University of Juiz de Fora*, Juiz de Fora, Brazil
¶ *Graduate Program in Electrical Engineering*, *Federal University of Juiz de Fora*, Juiz de Fora, Brazil
‖ *Department of Computer Science*, *Federal University of Minas Gerais*, Belo Horizonte, Brazil
** *Department of Mechanical Engineering*, *Federal University of Juiz de Fora*, Juiz de Fora, Brazil
Emails: {camandaroba.cristianogregory, sophia.werneck, caian.jesus, malta.matheus}@estudante.ufjf.br, jzibetti@ufmg.br, fredericoguimaraes@ufmg.br, eduardo.aguiar@ufjf.br

***Abstract*—This paper focuses on the development of a federated learning framework for predicting solar irradiance in photovoltaic power plants while preserving data privacy. The integration of decentralized machine learning techniques with two neural network architectures, Long Short-Term Memory (LSTM) and Multilayer Perceptron (MLP), is discussed and applied across multiple geographically distributed clients. Model evaluation was conducted to assess the predictive performance and robustness of the approaches. A comparative analysis of the effectiveness of LSTM and MLP models is presented, highlighting the superior capability of LSTM in capturing temporal dependencies inherent in solar irradiance data. Finally, the proposed federated learning approach demonstrates enhanced privacy protection and operational scalability for real-world solar energy systems. The reported results show that federated learning combined with LSTM provides a promising, privacy-preserving solution for accurate solar irradiance forecasting in distributed photovoltaic plants.**

***Index Terms*—Federated Learning, Solar Irradiance Forecasting, Time Series, Distributed Machine Learning.**

I. Introduction

Since the accelerated advancement of technology in recent years, there has been a profound transformation in the management and analysis of data. With the increasing volume of data generated by various industries, traditional computational techniques are being replaced by machine learning (ML) and artificial intelligence (AI) algorithms, which are capable of efficiently handling vast datasets [1], [2]. These technologies, particularly in fields such as healthcare, energy management, and telecommunications, play a central role in addressing complex challenges by analyzing and making predictions based on large-scale data. AI and ML models have become integral tools in decision-making, enabling the prediction of outcomes, optimization of processes, and enhancement of operational efficiencies. However, these models require not only vast amounts of data, but also sophisticated algorithms to process, interpret, and apply the insights generated. The success of AI and ML depends on their ability to continually learn from new data, making these fields increasingly dynamic and reliant on the effective management and protection of information.

In this context, data protection has become a critical concern due to these technological advancements. The importance of securing sensitive information, such as personal, financial, and operational data, is paramount, especially in the face of growing cybersecurity threats [3]. The risks associated with data breaches, unauthorized access, and data corruption can have severe consequences for both individuals and organizations. In particular, industries that rely heavily on data, such as the energy sector, must implement robust data security protocols to ensure the integrity and confidentiality of their operations. Solar power plants, for instance, manage critical data related to energy production, operational performance, and grid integration. Inadequate protection of this data can lead to significant operational risks, including unauthorized access to proprietary energy management systems or the exposure of confidential business information [4]. Therefore, ensuring the security of data within these systems is not only an ethical imperative but also a key factor in sustaining the credibility and profitability of organizations within the sector.

To address these challenges, this paper explores the integration of Federated Learning (FL) techniques to enhance the prediction of solar irradiance in photovoltaic (PV) plants while preserving data privacy [5]. FL, a decentralized ML technique, enables multiple institutions or systems to collaboratively train models without sharing sensitive data. This study presents a novel approach that combines FL with solar irradiance prediction, offering a promising solution to the dilemma of data security while improving the accuracy of predictions in solar energy systems. Through this process, it is possible to harness the benefits of collaborative ML without compromising data privacy.

The main contributions of this work are summarized as follows:

- Development of a federated learning framework for solar irradiance prediction across multiple clients while ensur-

ing data privacy.
- Application of LSTM (Long Short-Term Memory) and MLP (Multilayer Perceptron) models for predicting Global Horizontal Irradiance (GHI) in PV plants.
- Comprehensive evaluation and comparison of model performance using evaluation metrics to analyze the robustness and effectiveness of the models.

And our major conclusions are:

- FL proved to be an effective approach for predicting solar irradiance across multiple clients while maintaining data privacy, showcasing its potential in collaborative ML applications.
- Both LSTM and MLP models demonstrated promising results for GHI prediction, with LSTM models generally outperforming MLP models in capturing temporal dependencies and trends in the data.
- The evaluation metrics highlighted the strengths and limitations of each model, providing valuable insights into their robustness and effectiveness for solar energy forecasting in real-world scenarios.

The rest of the paper is organized as follows: Section II deals with the problem formulation. Section III discusses the methodology of the FL framework through the use of LSTM and MLP models. Section IV presents the computational experiments of the simulation. Section V states the main conclusions.

II. Problem Formulation

The exponential growth in data generation has necessitated stringent regulations to safeguard personal and operational information. Globally, frameworks such as the General Data Protection Regulation (GDPR) in the European Union [6], the California Consumer Privacy Act (CCPA) in the United States [7], and the General Data Protection Law (LGPD) in Brazil [8] have been established to ensure data privacy and security. These regulations mandate organizations to implement robust measures for data handling, emphasizing transparency, user consent, and the right to data access and deletion.

In the energy sector, particularly within PV plants, vast amounts of data are collected and processed to optimize performance and energy output. This data includes real-time information on solar irradiance, energy production metrics, and equipment status. However, the centralized storage and processing of such sensitive information pose significant risks. Data breaches can lead to unauthorized access, manipulation of operational parameters, and potential disruptions in energy supply.

Historically, several high-profile cyberattacks have underscored the vulnerabilities inherent in centralized energy systems. For instance, in December 2015, a coordinated cyberattack on Ukraine's power grid disrupted electricity supply to approximately 225,000 customers by compromising control systems and remotely opening circuit breakers at substations [9]. Similarly, in March 2019, a United States utility experienced a cyber intrusion that exploited a known vulnerability in unpatched Cisco firewalls, causing intermittent disruptions to Supervisory Control and Data Acquisition (SCADA) communications across multiple remote power generation sites, including PV and wind installations [10]. These incidents highlight the critical need for robust cybersecurity measures and the potential benefits of decentralized data processing techniques.

To mitigate these challenges, federated learning (FL) has emerged as a promising solution. FL enables the development of ML models across decentralized devices or servers holding local data samples, without exchanging them [11]. This strategy not only enhances data privacy but also reduces the risk of data breaches associated with centralized data storage. In the context of PV power plants, FL can facilitate accurate solar irradiance forecasting by leveraging data from multiple sources while ensuring that sensitive information remains localized.

Moreover, FL enables collaboration among clients that may be competitors in the same market, such as different energy companies operating PV plants. Since the data remain private and only model updates are shared, organizations can jointly train federated models to improve predictive accuracy without revealing proprietary or sensitive information. This capability allows rival companies to benefit mutually from shared learning, fostering cooperation while preserving competitive boundaries.

Under this perspective, the implementation of federated learning in PV power plants is essential, especially when predictive maintenance is concerned, in order to:

- Enhance solar irradiance predictions by utilizing several datasets from multiple sources of different companies.
- Strengthen data security by minimizing data transfer and reducing the risk of centralized data breaches.
- Improve operational efficiency through collaborative model training while adhering to stringent data protection regulations.

Taking into account the scenario previously exposed in this section, this work aims to develop a FL framework for solar irradiance prediction across multiple clients, ensuring data privacy and enhancing the robustness and effectiveness of the predictive models in real-world photovoltaic power plant scenarios.

III. Methodology

The methodology adopted for the development of the prediction model, which is the focus of this study, is based on the concept of FL, integrating it with Artificial Neural Networks (ANNs). FL will be explored in detail in this section, along with the other techniques used to build the model.

FL is a field of distributed ML aimed at training multiple models collaboratively, where the data remains localized at its original sources, and only model updates are shared between the clients [12]. This technique is particularly useful in scenarios where the data is geographically distributed, and centralizing it would be infeasible or undesirable due to privacy concerns or regulations [13]. The process starts with the sending of a generic model to all participating devices,

each of which locally trains the model using its own data. After local training, the models are sent to a central server, where the weight updates are aggregated to form a global model. This global model is then redistributed to the participating devices, which use it to continue training, and the cycle is repeated until the model reaches satisfactory performance.

The use of FL in this scenario allows the model to learn collaboratively from the data of different solar plants or meteorological stations without compromising the privacy of local data. Additionally, federated learning provides greater computational efficiency since training is distributed among the participants, and only model updates are communicated between the parties. This approach also enables scalability, allowing more devices to be integrated into the training process as the network of solar plants or meteorological stations grows.

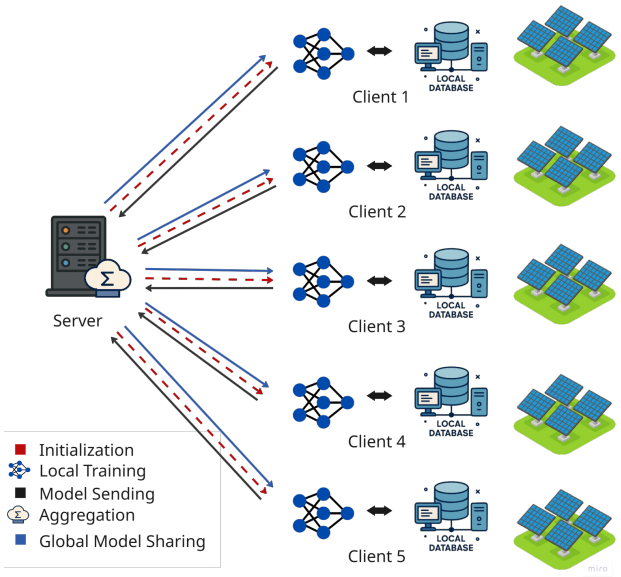

Fig. 1. Illustration of Federated Learning

The main advantage of FL is privacy preservation, as the data remains localized and is not directly shared. This is especially important when dealing with sensitive data, such as meteorological information or operational data from solar plants. Furthermore, FL allows the model to be trained across a network of devices, each contributing a part of the data, resulting in a more generalized model. The figure 1 illustrates the fundamentals steps that constitute the training procedure in FL, which are explained in detail below.

Step 1: Client Initialization

A global model is initialized on the central server and then sent to all clients, serving as the starting point for local training.

1) **Preprocessing**

Each client uses a dataset containing hourly GHI measurements from a specific Brazilian photovoltaic plant. However, the raw data were originally sampled at 4-minute intervals and contained missing values due to sensor failures or communication losses. To address this, the Solcast Historic API was used to retrieve estimated GHI values and fill the identified gaps. Solcast provides irradiance data based on a combination of geostationary satellite imagery and radiative transfer models, considering variables such as solar position, cloud coverage and thickness, aerosol content, and local atmospheric conditions. These satellite-based estimates, although not direct ground measurements, offer reliable approximations widely used in the solar industry for forecasting and historical reconstruction.

Missing intervals were identified programmatically, and only meaningful daytime gaps were considered for imputation. Once merged, the complete datasets were resampled to a 1-hour resolution, and negative irradiance values, which may result from sensor errors, were clipped to zero to ensure physical plausibility.

Following the completion of data processing, the next step is data splitting, where the data are separated into training and testing sets, with an 80% and 20% division, respectively. After the split, normalization is performed using only the training set samples to define the scaler, in order to prevent the test set values from influencing, even indirectly, the training data. Defining this training time series as $\mathbf{Q} = \{q_1, q_2, \ldots, q_j\}$, it is possible to define $q_{\max}$ and $q_{\min}$ as the minimum and maximum values that constitute the data in $\mathbf{Q}$. Additionally, q_k is defined as the k-th sample and $\bar{q}_k$ as the normalized value of q_k. Therefore, the following equation is obtained:

$$\bar{q}_k = \frac{q_k - q_{\min}}{q_{\max} - q_{\min}} \tag{1}$$

2) **Feature Selection**

The selection of inputs for the model was based on autocorrelation and partial autocorrelation analyses. As illustrated in Figure 2, the first-order and twenty-fourth-order samples exhibited the highest correlation coefficients, and also visually reflected a seasonal pattern characteristic of the data used [14], [15]. This seasonal pattern served as a relevant criterion in selecting these orders as inputs for the model.

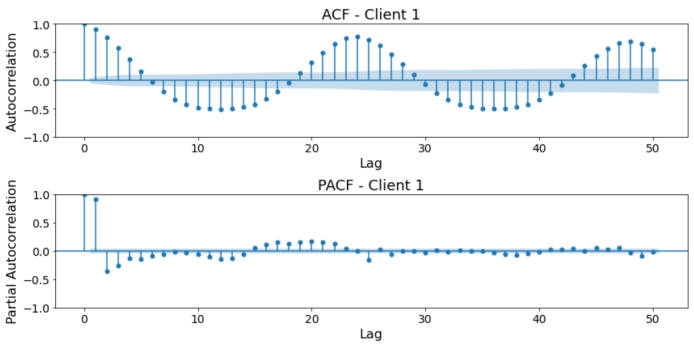

Fig. 2. Autocorrelation and Partial Autocorrelation of Client 1

Additionally, to enhance the short-term trend perception of the time series and mitigate the effects of noise present in the data, the use of moving averages was adopted, applied to the same orders [16]. Thus, the model inputs consisted of the following elements: the value of the last sample, the twenty-fourth previous sample, the last moving average sample, the twenty-fourth moving average sample. This configuration was designed to provide information that describes the behavior of the time series, incorporating not only the original values but also the moving averages, allowing the model to discern behavioral patterns over time.

3) **Model Architectures**

To build the model trained locally via FL, two distinct architectures were employed, MLP and LSTM. The selection of these architectures was motivated by their inherent characteristics and established performance in time series forecasting and regression tasks. The MLP, a fundamental feedforward neural network, was chosen for its proven capability in capturing complex non-linear relationships within data [17].

Further details on MLP and artificial neural networks can be found in [18]. The main equation representing the connections in the MLP is:

$$f_{i,j}(x) = \sigma\left(\sum_{k=1}^{n_i} \omega_{jk}^{i} x_k - \theta_j^i\right), \tag{2}$$

where σ denotes the activation function, which in this case is the ReLU, ω_{jk}^{i} denotes the weight, x_k is the input value and θ_j^i represents the bias term.

The LSTM networks constitute a specialized class of Recurrent Neural Networks (RNNs) designed to effectively learn long-range dependencies in sequential data [19]. They are especially suited for temporal forecasting applications, such as modeling energy consumption. Unlike conventional RNNs, which often struggle with the vanishing gradient problem when attempting to capture distant temporal information, LSTMs feature an internal architecture that facilitates the preservation and manipulation of important information over extended periods.

Each LSTM unit consists of multiple gates that regulate the flow of information within the cell. The main gates, namely the forget gate, the input gate and the output gate, control, respectively, the decision on which previous information should be discarded, which new information should be incorporated into the internal state, and which data should be emitted as output for the next step in the sequence.

- **Forget Gate (f_t):** This gate is responsible for determining which parts of the previous cell state (C_{t-1}) will be retained or discarded. The decision is based on the current input (x_t) and the previous hidden state (h_{t-1}), which are passed through a sigmoid activation function, as shown in Equation 3. Values close to 1 indicate that the corresponding information should be preserved, whereas values near 0 lead to the information being forgotten.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$

- **Input Gate (i_t):** It is responsible for determining the amount of new information to be incorporated into the cell state. This is achieved by working in conjunction with the Candidate State ($\tilde{C}_t$), which, through the use of the hyperbolic tangent activation function, serves as a candidate to update the cell's memory. The respective equations are:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5}$$

Thus, the Cell State (C_t) is updated based on the combination of the values obtained from the previous equations, resulting in:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{6}$$

- **Output Gate (o_t):** Decides which parts of the updated cell state are passed forward as the hidden state (h_t) and output of the LSTM unit.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = o_t \times \tanh(C_t) \tag{8}$$

Here, W and b represent the weights and bias learned by the model in the training, respectively. The gates work collaboratively, allowing the LSTM cell to selectively discard, update, and convey information across time steps, which contributes to its strong performance in sequential applications like time series forecasting.

Step 2: Local Training

Each solar plant locally trains the model received from the central server using its own data. The process of localized training ensures that the model captures the GHI patterns and behaviors specific to every solar plant.

Step 3: Model Sending

After completing round r, each client transmits the tuple

$$(\mathbf{w}_{k,r},\, n_k)$$

where $\mathbf{w}_{k,r}$ represents the local trained weights for the client k and n_k is the local sample count, used to weight its contribution on the subsequent averaging step.

Step 4: Aggregation

The central server consolidates the updates received from the selected clients $S_r \subseteq \{1, \ldots, K\}$ using the Federated Averaging (FedAvg) method [20]. At the end of round r, the

new global weight vector $\mathbf{w}_{r+1}$ is computed as a weighted average of the clients' local updates:

$$\mathbf{w}_r = \sum_{k \in S_r} \frac{n_k}{\sum_{j \in S_r} n_j} \mathbf{w}_{k,r}.$$

Here, S_r denotes the subset of clients participating in round r, n_k is the number of training samples at client k, and $\mathbf{w}_{k,r}$ is its locally updated weight vector after round r. The factor $\frac{n_k}{\sum_{j \in S_r} n_j}$ ensures each client's contribution to the global model is proportional to its data volume.

Step 5: Global Model Sharing

Finally, at the end of each communication round, the central server broadcasts the updated global model parameters (weight vector $\mathbf{w}_r$) back to all participating clients. This enables each client to synchronize its local model with the global parameters. The entire process is then repeated over multiple rounds, with the goal of progressively enhancing the overall performance of the global model.

Algorithm 1 FL for time-series GHI forecasting

Input: local datasets $\{D_k\}_{k=1}^{K}$
Output: final global federated model
Define model, number of communication rounds R, local epochs E, batch size B
for each client $k = 1, \ldots, K$ **do** ▷ Client side
    Load D_k; impute, resample, normalize
    Perform feature selection (ACF/PACF); construct inputs (x_k, y_k)
    Split into training/validation sets
    Initialize local model parameters $\mathbf{w}_{0,k}$
for $r = 1$ **to** R **do**
    for $k = 1$ **to** K **do** ▷ Client side, parallel
        Train local model on $(x_{\text{train}}, y_{\text{train}})$ for E epochs with batch size B
        Extract updated weights: $\mathbf{w}_{k,r}$
        Determine sample count: $n_k \leftarrow |x_{\text{train}}|$
        Send $\mathbf{w}_{k,r}$, n_k to server
    Server aggregates weights using FedAvg
    Broadcast updated $\mathbf{w}_r$ to clients
return final global model and performance report

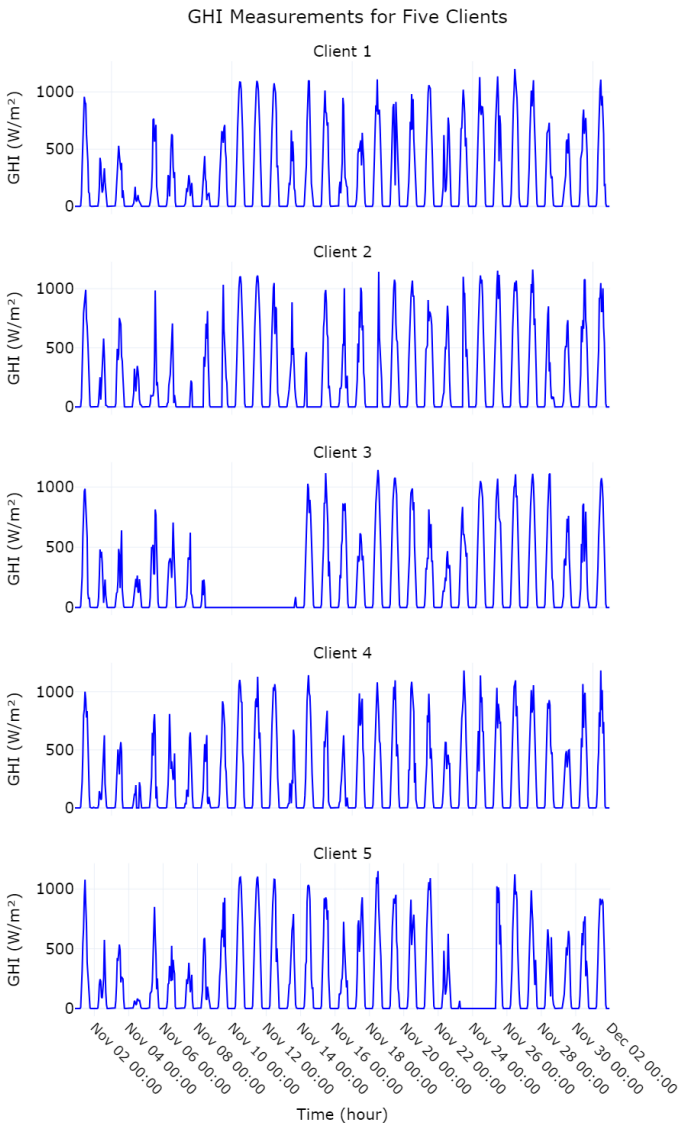

Fig. 3. Hourly GHI measurements for five clients, shown in separate subplots to illustrate temporal variations specific to each location.

IV. Computational Experiments

The algorithms employed to obtain the results presented in this section were implemented on a computer equipped with an AMD Ryzen 7 5700G processor operating at a clock frequency of 4.60 GHz, an NVIDIA RTX 4060 Ti graphics card, and 32 GB of RAM.

The datasets used in this work consist of data from five PV plants located in different regions of Brazil. The data utilized are GHI measurements, which were sampled on an hourly basis for the purpose of this project. The data spans from 9 p.m. on October 31, 2024, to 11 a.m. on January 10, 2025. The visual representation of a subset the datasets from the five PV plants is presented in Figure 3.

To assess the performance of the predictive models developed in this study, three widely recognized evaluation metrics were employed: Weighted Mean Absolute Percentage Error (WMAPE), Normalized Root Mean Square Error (NRMSE), and the Coefficient of Determination (R^2). These metrics provide comprehensive insights into the prediction and reliability of the models in forecasting GHI.

1) Weighted Mean Absolute Percentage Error (WMAPE): The WMAPE is a variation of the Mean Absolute Percentage Error (MAPE) that accounts for the scale of actual values, providing a more balanced error measurement, especially when dealing with data containing zero or near-zero values [21]. It is calculated as:

$$\text{WMAPE} = \frac{1}{\sum_{i=1}^{n} |y_i|} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{9}$$

where y_i represents the actual values, $\hat{y}_i$ denotes the predicted values, and n is the total number of observations.

2) Normalized Root Mean Square Error (NRMSE): The NRMSE provides a normalized measure of the differences between predicted and observed values, facilitating comparisons across different datasets or models [22]. It is defined as:

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{y_{\max} - y_{\min}} \tag{10}$$

where $y_{\max}$ and $y_{\min}$ are the maximum and minimum values of the actual data, respectively.

3) Coefficient of Determination (R^2): The R^2 metric quantifies the proportion of variance in the dependent variable that is predictable from the independent variables, serving as an indicator of the model's explanatory power [23]. It is computed as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{11}$$

where $\bar{y}$ is the mean of the actual values.

A. Results

All clients' datasets, as described in Section III, were preprocessed, normalized, and partitioned into training and testing sets with an 80:20 ratio. The training was conducted over 2 epochs and 20 rounds.

The MLP model was configured with two hidden layers, each comprising 68 neurons, utilizing the ReLU activation function. The LSTM model consisted of three hidden layers with 64, 32, and 16 neurons, respectively, and also employed the ReLU activation function in the dense layers. Both models were optimized using the SGD optimizer with a learning rate of 0.01.

FL was applied using the LSTM model across five clients. The datasets were partitioned into training and testing subsets, and the performance metrics, computed exclusively from daytime irradiance values in the test set, are summarized in Table I.

TABLE I
PERFORMANCE METRICS (WMAPE, NRMSE, AND R^2) FOR EACH CLIENT USING LSTM AND MLP MODELS

Client	Model	WMAPE (%)	NRMSE	R^2
1	LSTM	21.49	0.13	0.82
	MLP	24.98	0.14	0.79
2	LSTM	18.57	0.12	0.85
	MLP	19.74	0.12	0.84
3	LSTM	12.84	0.08	0.93
	MLP	21.87	0.12	0.85
4	LSTM	24.55	0.09	0.89
	MLP	22.42	0.12	0.83
5	LSTM	21.50	0.13	0.82
	MLP	22.71	0.14	0.81

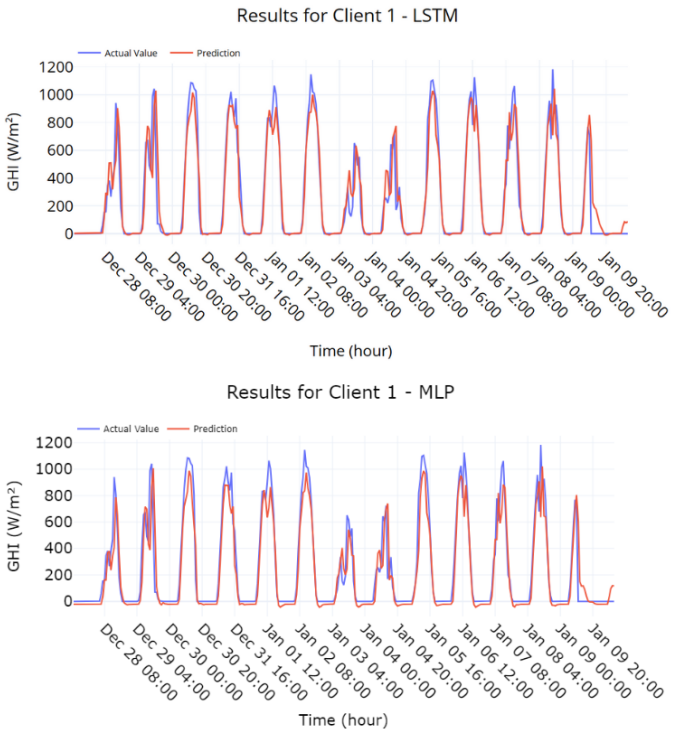

Fig. 4. Predicted and actual GHI values for client 1 using LSTM and MLP models.

Figure 4 illustrates the predicted GHI values over time for client 1, comparing the outputs from both the MLP and LSTM models against the actual observed data. As illustrated, both models demonstrate comparable performance, closely following the temporal variations in the real measurements. Although minor differences exist, neither model shows a clearly superior fit in this particular case, indicating that both architectures are capable of effectively modeling the irradiance patterns for this client.

B. Comparative Analysis

Considering the results presented, it can be observed that both the LSTM and MLP models, trained within a FL framework, were capable of adequately capturing variations in solar irradiance over time. However, the LSTM model consistently outperformed the MLP across most evaluation metrics applied to the five clients. This advantage is primarily attributed to the LSTM's ability to retain temporal dependencies, which is a critical requirement when dealing with time series data such as GHI. In contrast, the MLP is limited in modeling sequential patterns, which affects its predictive performance in more dynamic scenarios.

This approach, using FL, facilitated the integration of diverse local data patterns without centralized data aggregation, which improved the models' generalization capabilities. Moreover, federated learning helped mitigate the effects of data heterogeneity among clients, reducing overfitting risks and enhancing the stability and robustness of the forecasts. Con-

sequently, the models, especially the LSTM, benefitted from a richer and more varied training experience that contributed to better performance in real-world, distributed solar irradiance forecasting scenarios.

Analyzing the evaluation metrics, the LSTM model achieved lower WMAPE and NRMSE values and higher R^2 scores for the majority of clients. Although the MLP also delivered satisfactory results in some cases, the LSTM demonstrated superior adaptability to complex temporal dynamics, reinforcing its robustness for forecasting in distributed solar energy systems.

V. Conclusions

This work proposed a FL framework for predicting solar irradiance in PV power plants using distributed datasets from multiple clients while preserving data privacy. Two neural network architectures were evaluated: MLP and LSTM. The results demonstrated that both models are capable of learning relevant patterns in irradiance data. Nevertheless, the LSTM consistently outperformed the MLP in most scenarios. This performance can be attributed to the LSTM's capacity to capture temporal dependencies, which are fundamental in time series forecasting tasks such as GHI prediction.

From a practical perspective, the integration of FL with LSTM models provides an effective and privacy-preserving solution for energy management in distributed PV systems. By enabling collaborative model training without requiring direct data sharing, the proposed approach addresses critical concerns regarding data security and regulatory compliance. Furthermore, it enhances operational efficiency and supports informed decision-making in real-world applications. These findings highlight the potential of FL to be applied in other domains where data sensitivity and decentralization are key constraints, suggesting promising directions for future research and development.

Acknowledgment

The authors would like to acknowledge the National Council for Scientific and Technological Development — CNPq (process 317939/2023-8), Minas Gerais State Research Support Foundation — FAPEMIG (APQ-00642-24) and Federal University of Juiz de Fora — UFJF, National Council for Scientific and Technological Development (CNPq), Grant no. 304856/2025-8, "*Aprendizado de Máquina Colaborativo e Proteção à Privacidade*"; Coordination for the Improvement of Higher Education Personnel (CAPES) through the Academic Excellence Program (PROEX) for financial support.

Frederico G. Guimarães and Julia Z. C. Mellos acknowledge partial support from Kunumi and Embrapii, project PDCC-2412.0030.

References

[1] Soori, M., Arezoo, B., Dastres, R., "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," Cognitive Robotics, vol. 3, pp. 54–70, 2023, doi: 10.1016/j.cogr.2023.04.001.

[2] L'Heureux, A., Grolinger, K., ElYamany, H., Capretz, M., "Machine Learning With Big Data: Challenges and Approaches, "IEEE Access, vol. 99, pp. 1–1, 2017, doi: 10.1109/ACCESS.2017.2696365.

[3] Saeed, S., Altamimi, S., Alkayyal, N., Alshehri, E., Alabbad, D.,"Digital Transformation and Cybersecurity Challenges for Businesses Resilience: Issues and Recommendations," Sensors, vol. 23, Jul. 2023, doi: 10.3390/s23156666.

[4] Govea, J., Gaibor-Naranjo, W., Villegas-Ch, "Transforming Cybersecurity into Critical Energy Infrastructure: A Study on the Effectiveness of Artificial Intelligence ," Systems, 12(5), 165, May, 2024, doi: 10.3390/systems12050165.

[5] Reddy, G. and Kumar, Y., "A Beginner's Guide to Federated Learning," 2023 Intelligent Methods, Systems, and Applications (IMSA), July, 2023, pp. 557-562, doi: 10.1109/IMSA58542.2023.10217383.

[6] European Union, Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016, Official Journal of the European Union, L 119, pp. 1–88, May 4, 2016.

[7] State of California, "California Consumer Privacy Act (CCPA), California Civil Code §§ 1798.100–1798.199," enacted June 28, 2018.

[8] Brazil, "Lei Geral de Proteção de Dados Pessoais (LGPD), Law No. 13.709," enacted August 14, 2018.

[9] K. Zetter. "Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid." wired.com. Accessed: May. 18, 2025. [Online.] Available: https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/

[10] Harrou, F., Taghezouit, B., Bouyeddou B. and Sun, Y., "Cybersecurity of photovoltaic systems: challenges, threats, and mitigation strategies: a short survey," Frontiers in Energy Research, vol. 11, Sep. 2023, doi: 10.3389/fenrg.2023.1274451

[11] Daly, K., Eichner, H., Kairouz, P., McMahan, H. B. and Ramage, D., Xu, Z., "Federated Learning in Practice: Reflections and Projections," IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA), pp. 148–156, Oct. 2024. doi: 10.1109/TPS-ISA62245.2024.00026

[12] Yurdem, B., Kuzlu, M., Gullu, M. K., Catak, F. O., and Tabassum, M., "Federated learning: Overview, strategies, applications, tools and future directions," *Heliyon*, vol. 10, no. 19, article e38137, 2024, doi: 10.1016/j.heliyon.2024.e38137.

[13] Sah, D. K., Vahabi, M., and Fotouhi, H., "Federated learning at the edge in Industrial Internet of Things: A review," *Sustainable Computing: Informatics and Systems*, vol. 46, article 101087, 2025, doi: 10.1016/j.suscom.2025.101087.

[14] Fiveable, "9.2 Identifying seasonal patterns in ACF and PACF – Intro to Time Series," edited by B. Bahr, Fiveable, 2024. [Online]. Available: https://library.fiveable.me/intro-time-series/unit-9/identifying-seasonal-patterns-acf-pacf/study-guide/k0imQyzItSW3JUuc. [Accessed: May 21, 2025].

[15] Tinungki, G. M., "The analysis of partial autocorrelation function in predicting maximum wind speed," *IOP Conference Series: Earth and Environmental Science*, vol. 235, no. 1, article 012097, Feb. 2019, doi: 10.1088/1755-1315/235/1/012097.

[16] Su, Y., Cui, C., and Qu, H., "Self-attentive moving average for time series prediction," *Applied Sciences*, vol. 12, no. 7, article no. 3602, 2022, doi: 10.3390/app12073602.

[17] Abdurrakhman, A., Sutiarso, L., Ainuri, M., Ushada, M., and Islam, M. P., "A Multilayer Perceptron Feedforward Neural Network and Particle Swarm Optimization Algorithm for Optimizing Biogas Production," Energies, vol. 18, no. 4, art. no. 1002, 2025, doi: 10.3390/en18041002.

[18] Yu, Y. and Zhang, Y., "Multi-layer Perceptron Trainability Explained via Variability," arXiv preprint arXiv:2105.08911, 2023. [Online]. Available: https://arxiv.org/abs/2105.08911

[19] Lindemann B., Muller, T., Vietz, H., Jazdi, N., Weyrich, M., "A survey on long short-term memory networks for time series prediction," Procedia CIRP, vol. 89, pp. 650–655, 2021, doi: 10.1016/j.procir.2021.03.088

[20] Qi, P., Chiaro, D., Guzzo, A., Ianni, M., Fortino, G., and Piccialli, F., "Model aggregation techniques in federated learning: A comprehensive survey," Future Generation Computer Systems, vol. 150, pp. 272–293, 2024, doi: 10.1016/j.future.2023.09.008.

[21] V. Kothari. "Time Series Evaluation Metrics — MAPE vs WMAPE vs SMAPE — which one to use, why and when? — Part1." Medium.com. Accessed: May. 18, 2025. [Online.] Available: https://medium.com/@vinitkothari.24/time-series-evaluation-metrics-mape-vs-wmape-vs-smape-which-one-to-use-why-and-when-part1-32d3852b4779

[22] Botchkarev, A., "Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Ty-

pology," Interdisciplinary Journal of Information, Knowledge, and Management, vol. 14, pp. 45–79, Sep. 2018. doi: 10.28945/4184

[23] Chicco, D., Warrens, M. J. and Jurman, G., "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Computer Science, vol. 7, pp. e623, Jul. 2021. doi: 10.7717/peerj-cs.623