Energy and AI 18 (2024) 100452

Contents lists available at ScienceDirect

Energy and AI

journal homepage: www.elsevier.com/locate/egyai

Enhancing PV feed-in power forecasting through federated learning with differential privacy using LSTM and GRU

Pascal Riedel [a,*], Kaouther Belkilani [b], Manfred Reichert [a], Gerd Heilscher [b], Reinhold von Schwerin [a]

[a] Institute of Databases and Information Systems, Ulm University, Ulm, 89081, Germany
[b] Smart Grids Research Group, Ulm University of Applied Sciences, Ulm, 89081, Germany

HIGHLIGHTS

- Federated learning with long short-term memories and gated recurrent units for electrical feed-in power forecasting.
- Training on energy data from real residential households with PV-systems connected to the low-voltage grid.
- Proposing a federated-driven method with differential privacy for the privacy-preserving prediction of the feed-in power.
- Advanced federated aggregation strategies to mitigate adverse data distributions on the model performance.
- Model performance comparison and analysis of different training methods.

ARTICLE INFO

Keywords:
Federated learning
Deep learning
Recurrent neural networks
Data privacy
Solar power forecasting
Smart grid
Residential photovoltaic

ABSTRACT

Given the inherent fluctuation of photovoltaic (PV) generation, accurately forecasting solar power output and grid feed-in is crucial for optimizing grid operations. Data-driven methods facilitate efficient supply and demand management in smart grids, but predicting solar power remains challenging due to weather dependence and data privacy restrictions. Traditional deep learning (DL) approaches require access to centralized training data, leading to security and privacy risks. To navigate these challenges, this study utilizes federated learning (FL) to forecast feed-in power for the low-voltage grid. We propose a bottom-up, privacy-preserving prediction method using differential privacy (DP) to enhance data privacy for energy analytics on the customer side. This study aims at proving the viability of an enhanced FL approach by employing three years of meter data from three residential PV systems installed in a southern city of Germany, incorporating irradiance weather data for accurate PV power generation predictions. For the experiments, the DL models long short-term memory (LSTM) and gated recurrent unit (GRU) are federated and integrated with DP. Consequently, federated LSTM and GRU models are compared with centralized and local baseline models using rolling 5-fold cross-validation to evaluate their respective performances. By leveraging advanced FL algorithms such as FedYogi and FedAdam, we propose a method that not only predicts sequential energy data with high accuracy, achieving an R^2 of 97.68%, but also adheres to stringent privacy standards, offering a scalable solution for the challenges of smart grids analytics, thus clearly showing that the proposed approach is promising and worth being pursued further.

1. Introduction

In Germany [1], the majority of photovoltaic (PV) systems is installed in residential areas and connected to low-voltage distribution electricity grids [2,3]. A residential PV system is characterized as a PV installation possessing a maximum rated power capacity not exceeding 10 kW [2,4–6]. The electricity produced from residential PV systems for 12% of total net electricity generation in Germany, in the year 2023 alone [7]. It is estimated that gross electricity generation in Germany from 100% renewable energies will increase to up to 780 TWh by 2050 [8]. Thus, a major increase in the number of PV systems has been observed in recent years, introducing new challenges for the electricity grid management and control in low-voltage grids. These

* Corresponding author.
E-mail addresses: pascal.riedel@uni-ulm.de (P. Riedel), kaouther.belkilani@thu.de (K. Belkilani), manfred.reichert@uni-ulm.de (M. Reichert), Gerd.Heilscher@thu.de (G. Heilscher), Reinhold.vonSchwerin@thu.de (R. von Schwerin).

https://doi.org/10.1016/j.egyai.2024.100452
Received 23 August 2024; Received in revised form 30 October 2024; Accepted 16 November 2024
Available online 23 November 2024

challenges encompass power quality, voltage stability, and reactive power support management, all related to the integration of PV systems into low-voltage grids [2,9,10]. The objective of such an electrical power system is the reliable and economical supply of electric power to end-customers [2,11].

For the solar power prediction from PV systems, measurement techniques can be categorized as physical or statistical. However, in practice, the lines between these approaches are often unclear. For example, numerical weather prediction models or sky images are used in physical approaches as a part of irradiance prediction, whereas statistical approaches forecast solar irradiance from statistically derived past data [12]. In deterministic approaches, output prediction is done by using PV device models obtained through different software such as PVSyst, and system advisor model (SAM), among others [13]. Sometimes these prediction methods are not able to reflect the variations in the data. Due to the prevalence of such cases, probabilistic or machine learning (ML) models are commonly used [10,14].

Distribution system operators (DSOs) need to plan, operate, and maintain the electrical grid to avoid voltage band violations or overloading of grid assets [3,11]. Thus, the inherent objective of a DSO is to avoid unnecessary financial investments in brute force grid reinforcement which may occur due to missing knowledge on the PV power contribution in different grid sections. The DSOs require accurate information on distributed energy resources in the electrical grid in order to choose appropriate grid operations. Power flow from the residential PV has a tremendous effect on the load flow of the low voltage distribution grid transformer. By accurately predicting power flow from residential PV systems, grid operators can better anticipate fluctuations in supply and demand, this can help mitigate issues such as voltage fluctuations, overloading of transformers, and potential grid instability.

Data-driven methods applied on end-user meter data from residential PV systems for feed-in power forecasting can support the grid management of the DSOs to increase the stability and reliability of the electrical grid, leveraging cost-saving effects [3,10,12]. End-user data streams are historical and include past feed-in powers and residual loads in a time series format [14,15]. Historical weather information can be added on top of the training data to increase the data quality. The end-user based feed-in forecast can serve as a valuable input parameter for the DSO for the active control and planning process on the low-voltage grid. The top-down structure and the unidirectional power flow of conventional distribution grids allow operating parameters such as voltage limits or resource utilization to be maintained to a large extent during grid planning. This simplifies operational management, as extensive monitoring of the system status is not necessary. The addition of PV systems, which mainly takes place in the distribution grid due to the size of the systems, changes this top-down structure and thus also increases the complexity of grid operation. This leads to bidirectional current flows and makes it difficult to maintain the operating parameters. A feed-in forecast therefore plays a major role for the DSOs. The responsibility of a DSO is also shown in Fig. 1.

A key challenge for feed-in power forecasting is data privacy. Since forecasting at low-voltage level, typically means at the end-customer level, the general data protection regulation (GDPR) must be considered within Europe [16,17]. However, ML and statistical forecasting models usually require access to aggregated centralized training data. This approach harbors data privacy risks as well as a risk of a GDPR breach. Federated learning (FL) with differential privacy (DP) is proposed in this study to predict the amount of energy fed into the grid in a privacy-preserving manner and to avoid the potential risks stemming from data centralization. Utilizing FL, deep learning (DL) models, including neural networks, can be trained using distributed datasets without necessitating the aggregation and centralized storage of this data [18–22]. In fact, the data remains in *data silos* (e.g. clients on smart meter gateways) and only locally trained model weights are sent to a secure server for a model aggregation. The proposed prediction method in this study incorporates FL models with DP, a privacy technique that

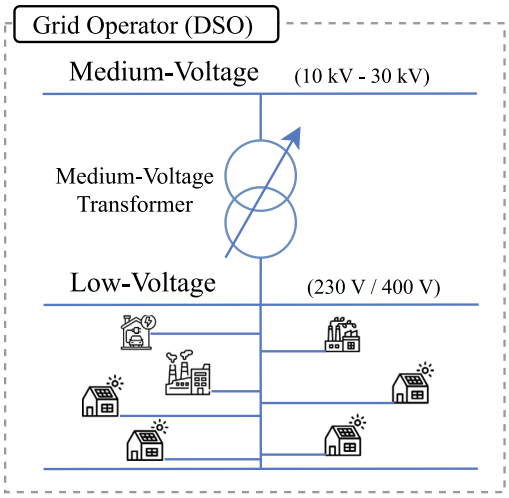

Fig. 1. Electricity from the medium-voltage grid has to be converted into the low-voltage grid for use by end customer (e.g. households or small factories) Generation structure and power flows in a grid with renewable electricity generation. The DSO coordinates energy distribution and demand management for both grids.

injects noises into the federated model [19,20,23], to fulfill the data privacy and security requirements for data-driven analyses in low-voltage grids. The dataset for the proposed prediction method encompasses authentic meter data from residential PV systems in Southern Germany, marking the first instance of its application in an FL context.

Another challenge is the non-independently and identically distributed (Non-IID) data between the data silos in an FL system [24]. It poses a challenge to FL by influencing the federated model performance [25]. Depending on the number of data silos and the level of Non-IID between them, the federated model may have difficulties to find an optimal local minima resulting in higher error rates in the model inference task. Adding a new data silo in an existing FL system can easily disturb the model convergence when the joining training data is heavily Non-IID [24–27]. This requires the development and implementation of advanced federated aggregation strategies designed to mitigate the adverse effects of Non-IID data, ensuring that the federated model remains effective and robust across all data silos. To reduce the impact of Non-IID, advanced FL aggregation strategies such as FedAdam [24] and FedYogi [26] are used in this study.

Concerning the ML model architectures, it has been shown by various prior studies that sequential DL models yield promising outcomes when applied to time series energy data [15,28–32]. Consequently, in this study, federated long short-term memory (LSTM) and gated recurrent unit (GRU) models, trained on meter data from PV systems, are evaluated in comparison to their centralized counterparts. The findings from these experiments provide valuable insights into FL systems with sequential DL models on distributed energy data, which can support the DSO in optimizing the low-voltage grid management.

In this comparative study, real-world measurements of PV solar power generation from individual households, with a time granularity of 15 min, are utilized alongside regional solar irradiance data.

Thus, the highlights of this study can be summarized as follows:

- Accurate predictions of the feed-in power into the grid with a time resolution of 15 min per sample by different data-driven forecasting methods.
- Enhancing the required data privacy by the DSO for smart grid analytics using FL with DP.
- Training federated LSTM and GRU models directly on prepared real-world meter data, including regional solar irradiance data, using various federated aggregation strategies across different test settings, thereby facilitating a fair comparison in terms of accuracy.

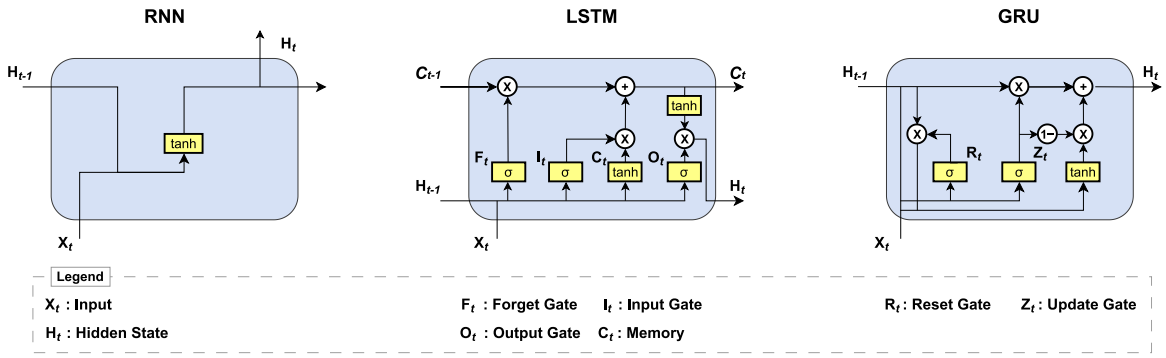

Fig. 2. Sequential neural network model architectures.

- Comparing the federated models with local and centralized baseline models through a systematic evaluation design.

The structure of the remainder of this paper is organized as follows: Related work, the DL architectures employed, and the concept of FL are delineated in Section 2. The dataset used to train the models and the characteristics of the data as well as the necessary pre-processing steps are elaborated in Section 3. Also presented within that section are the different training approaches, along with the workflow of the proposed feed-in power forecasting method employing FL and DP. The experimental settings, including the hyperparameter search and testing settings for both approaches, is described in Section 4. The experimental results derived from comparative tests and the implications of these findings for low-voltage grid management are described in Section 5. From these results, the most suitable privacy-preserving prediction method for end-user feed-in power forecasting is selected. A conclusion of the main findings of this paper is provided in Section 6.

2. Related work

In this section, previously published studies and works that have addressed the topic of data-driven feed-in power forecasting are discussed. Initially, an explanation is provided for the neural network model architectures that are used for the feed-in power forecasting, with a focus on sequential DL models. Subsequently, an outline is presented for the state of the art research concerning FL, DP and relevant federated model aggregation strategies.

2.1. Model architectures

Several papers demonstrated that a data-driven forecasting is best achieved with sequential neural network architectures trained on sufficient time series data [15,27,29,30,33–36]. In this context, the LSTM model is often discussed. Introduced by Hochreiter and Schmidhuber [37] the LSTM model uses a recurrent neural network (RNN) structure and considers temporal dependencies within the data, enabling the learning of seasonal patterns that are a common characteristic of time series. In addition, the LSTM models showed they are successfully mitigating the vanishing gradient problem, a common challenge in DL and RNNs in particular where the gradients in the training procedure exponentially decrease when they are propagated backwards through the network, resulting in a model with poor generalization [34,35,37,38]. As stated in [36,37,39] the forget, input and output gates in the LSTM model architecture manage the flow of information by selectively adding or removing information to the cell state, thus maintaining a longer memory compared to standard RNNs and preventing the vanishing gradient problem.

Jailani et al. [40] investigated the power of LSTM-based models in solar energy forecasting, comparing independent and hybrid LSTM models. The study highlighted the superiority of LSTM in forecasting solar radiation and generated PV power, showcasing LSTM's adaptability and effectiveness in different forecasting scenarios.

Skrobek et al. [41] demonstrated successfully LSTM models on the prediction of the sorption process in adsorption chillers. Their results indicate that the LSTM model was capturing the dynamics of sorption processes, showcasing the potential of LSTMs in optimizing cooling systems.

Kumar et al. [33] analyzed and compared LSTM models for forecasting solar and wind power in isolated microgrids, focusing on load frequency control. Their study addressed the stochastic nature of renewable sources and their integration into electrical power systems, using LSTM to predict the wind speed and solar irradiance.

A modern advancement of the LSTM architecture is the GRU architecture introduced by Cho et al. [42]. The GRU simplifies the LSTM architecture by combining the input and forget gates into a single update gate and merging the cell state and hidden state into one, thereby reducing model complexity. GRUs were shown in the literature to achieve comparable or superior performance to LSTMs in time series forecasting [28–31,39,42].

Kisvari et al. [43] utilized GRUs for accurate predictions of wind power. Their proposed GRU model not only offered faster training processes compared to LSTM but also showed less sensitivity to noises, which is important for the dynamic and unpredictable nature of wind power generation.

Mahjoub et al. [44] conducted a comparative study on the performance of GRU and LSTM models in forecasting electricity consumption based on power loads. Their research indicated that GRUs not only marginally surpassed LSTMs in terms of predictive accuracy but also demonstrated a considerable reduction for the model training time, especially when GRUs are applied on large-scaled datasets.

Skrobek et al. [45] compared LSTM, GRU and bidirectional LSTM models for predicting the mass of adsorption beds used in cooling systems. Although all three models performed similarly, the GRU model showed the highest accuracy in the prediction task, emphasizing the potential of GRU models in terms of accuracy and computational efficiency.

More traditional ML architectures such as random forests, decision trees, or support vector machines for feed-in power prediction are also discussed by [46–49], but several experiments have shown that they are outperformed by DL models [50,51]. Although [14,52] showed promising results when temporal convolutional networks are used, LSTMs and GRUs are generally better suited for the training on long seasonal time series data due to their inherent memory function and capability of capturing temporal dependencies.

Based on the findings of these papers, the LSTM and GRU models are federated and compared with baseline models in this study. The architectures of these models are depicted in Fig. 2.

2.2. Federated learning

The federated paradigm of training distributed DL models were discussed in some papers, addressing the concerns of missing data privacy

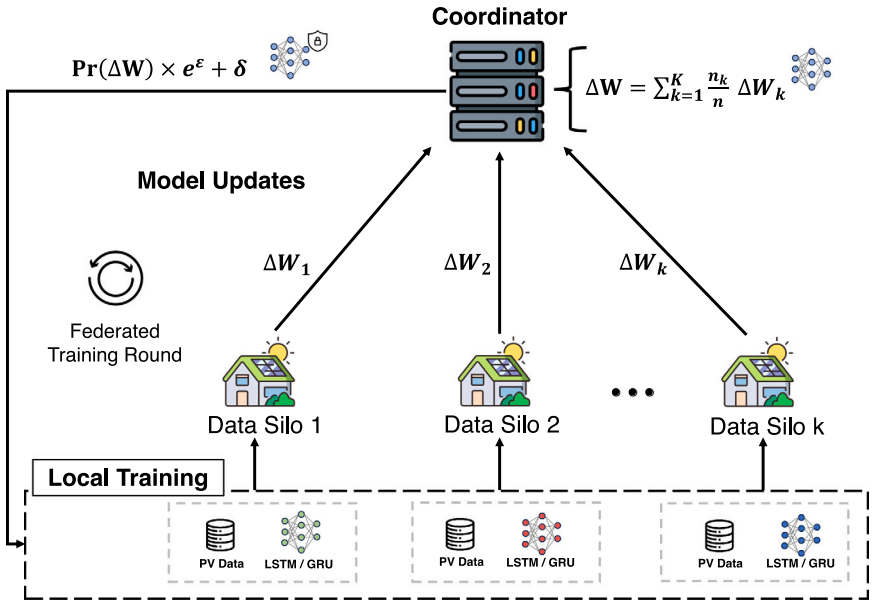

Fig. 3. Workflow of the privacy-enhanced federated feed-in forecasting with DP.

in centralized data-driven approaches [16,18,20,21,24,26,27,53–55]. In an FL system, training data remains within its data silos, thereby obviating the need for data centralization. A typical example of a data silo is a healthcare facility where GDPR compliance is mandatory due to sensitive patient data, e.g., for X-rays images [17]. After the local models are trained on each data silo, the raw model weights are transmitted and subsequently aggregated by a central coordinating server [16,21,26]. In this aggregation phase, all received model weights are combined by the coordinator in a node-wise averaging manner to build an improved single global model. This updated global model is then redistributed to all participating data silos and a new training round is started. For the global model convergence, the number of federated training rounds is set in advance, which is a tunable hyperparameter [54,56].

The authors of [18] characterize the federated training process as an optimization task defined as follows:

$$\min f(w) = \sum_{k=1}^{N} \frac{n_k}{n} F_k(w), \tag{1}$$

where N represents the number of data silos, n_k denotes the volume of sensitive training data on data silo k, n the total amount of training data across all silos and $F_k(w)$ refers to the local loss function of each silo. The updated local model weights are asynchronously sent back to the coordinator, where an updated global model is computed using an aggregation strategy such as federated averaging (FedAvg) [24,27] according to Eq. (1).

Widmer et al. [14] investigated FL and model personalization for electrical load forecasting. By benchmarking federated models against baseline centralized models, the researchers demonstrated that federated models can achieve performance and accuracy comparable to centralized models. The authors also introduced differential comparison, a method that compares the loss offsets from multiple data sources and considers different data constraints to provide a solution to the Non-IID problem in FL systems.

Zhang et al. [27] introduced a federated multi-energy load forecasting method using a modified LSTM model for optimizing microgrids. Their findings indicate that federated models can attain accuracy levels similar to those of centralized models, while also delivering higher precision compared to individual silo-based models.

Generally, FL can be utilized across diverse sectors. From the healthcare industry with patient data to autonomous driving with personal driving information, the necessary data protection aspects for ML in these industries can be addressed by FL [17,57]. Moreover, the federated approach also eliminates the cost-intensive process of centralizing data [18,26,54,55]. An overview of the complete FL process with FedAvg is visualized in Fig. 3.

2.3. Security in federated learning systems

In an FL system the training data in the silos remain privately within the local storage and are not moved to a central point (privacy by design), whereas traditional ML methods require centralized datasets. There are several enhanced security mechanisms to improve data privacy in FL that are being discussed in the research community.

Wu et al. [58], Zhou et al. [59], Zhu [60] all agree that secure multi-party computation (SMC) can enhance the security of FL systems. SMC protocols allow data silos to jointly compute functions on their net inputs while keeping those inputs private. This is achieved by obscuring individual data contributions of each silo, thus safeguarding the integrity of the overall model training process. Despite its benefits, SMC incurs substantial computational and communication overhead due to the requirement for secret sharing across data silos.

Zhu [60], Hussien et al. [61] propose the adoption of homomorphic encryption on model weights. This technique enables the performance of operations on encrypted data, thus allowing secure aggregation of model updates without exposing the underlying data or model parameters. It protects the integrity of the data and the privacy of the models' parameters across potentially insecure networks. Despite its potential, homomorphic encryption exhibits even greater computational complexity and further reduced processing speed compared to SMC, making it impractical for most of the real-world scenarios [62].

Riedel et al. [17], McMahan et al. [19], Tang et al. [55], Ouadrhiri and Abdelhadi [63] discuss the integration of DP to harden FL systems against privacy-specific attacks such as false data injection or membership inference attacks [64]. DP injects noises to the training data and model parameters, ensuring that the output of the training process does not reveal sensitive information about the training data [23]. This approach is crucial for maintaining confidentiality and trust of a data silo. The difficulty with DP comes with keeping the balance between privacy and model performance [63].

These enhanced security mechanisms increase security in FL systems, but also have their shortcomings. However, the inclusion of

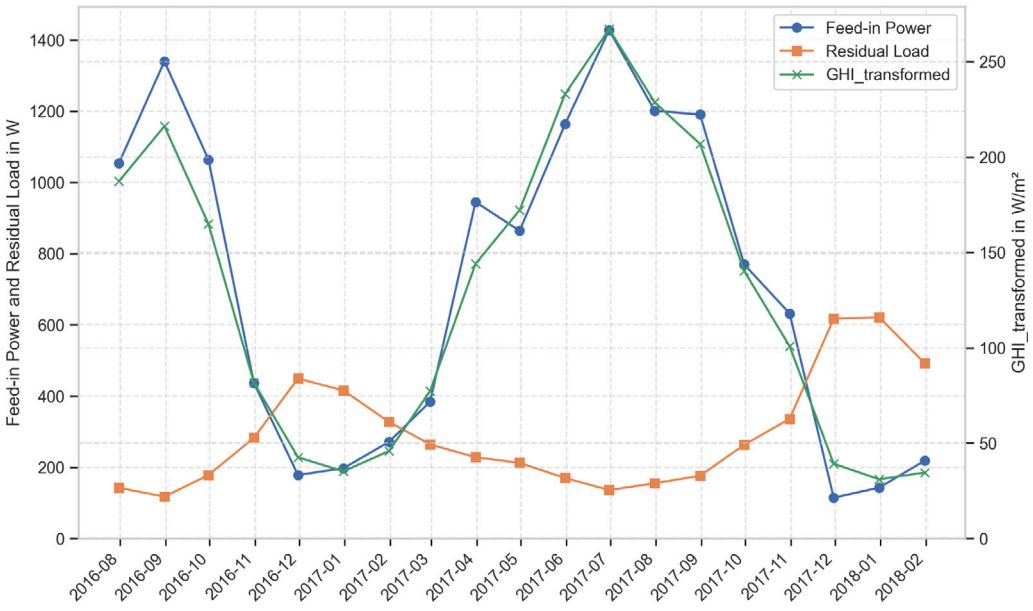

Fig. 4. Power metrics from a residential household with an installed PV system in the designated test area. Data points represent average values calculated over one-month intervals, illustrating seasonal variations and their impact on energy production and residual load.

DP has shown the most promising trade-off in some related works. Therefore, DP is utilized for the proposed forecasting method in this study.

2.4. Model aggregation strategies

Extensions to the FedAvg algorithm (see Eq. (1)) were proposed in other papers, underscoring the necessity for more robust solutions to address the challenges posed by Non-IID data silos and domain shifts in FL systems.

FedNova, as proposed by [65], introduces a normalized variant of the FedAvg algorithm. This approach facilitates accelerated model convergence by mitigating objective inconsistencies arising from heterogeneous local model updates, which otherwise can result in suboptimal stationary points of the global objective function.

Karimireddy et al. [66] suggested the utilization of stochastic control variables during the model aggregation phase to mitigate client drift, which arises as a direct consequence of the presence of heavily Non-IID data across silos. Their approach aims to enhance the convergence stability and efficiency of FL systems by aligning local updates more closely with the global model objective.

Reddi et al. [24] introduced several extensions of FedAvg to optimize FL in non-convex settings. Drawing from advancements in centralized DL, they proposed FedAdam, FedAdagrad, and FedYogi. These aggregation strategies use pseudo-gradients and accommodate negative values, thereby enhancing convergence rates and model robustness compared to the standard FedAvg algorithm.

Zhang et al. [27] examined FL on microgrid data with the four advanced model aggregation strategies FedAvg, FedAdagrad, FedAdam, and FedYogi. The authors also examined the security aspect of these strategies in training FL models. They demonstrated the effectiveness of adaptive optimization techniques for suboptimal data distributions and under the assumption of data injection attacks. Their results showed that FedAdagrad can maintain stability and has the best prediction performance. However, Riedel et al. [26], Reddi et al. [24] showed that FedAdagrad generally performs worse than FedAdam and FedYogi.

Considering these findings, LSTM networks and GRUs are trained for the task of feed-in power forecasting using FedAvg, FedAdam, and FedYogi in this study.

3. Methods

In this section the dataset, data preparation process and the different model training approaches for conducting the comparison study are delineated.

3.1. Dataset

For this paper, meter data from a suburban residential area in the southern German city of Ulm is used. The dataset comprises feed-in power time series generated by three distinct PV systems installed in households within the test area, spanning a period of two to three years. Additionally, the dataset encompasses residual load data from the DSO to address energy shortages, which may have arisen due to suboptimal PV power generation. Both variables were measured with a time resolution of 15 min. Fig. 4 shows the monthly averaged measured feed-in power of a single PV system and residual load of a household from the test area. As can be seen in Fig. 4, the feed-in power is higher in the summer months than in the winter months, clearly demonstrating the seasonality in the time series dataset.

To include meteorological data, time series of solar surface irradiance over the same period and with the same time resolution are used to create a more stable forecasting model. The solar irradiance data for the test area were obtained from Solargis including global horizontal irradiance (GHI) values in W/m^2 from January 1, 2015, to December 31, 2018, offering high-resolution meteorological information. It includes spatially disaggregated information derived from the ERA5 model, using Solargis' proprietary methods to achieve a spatial resolution of 250 m for solar data and 1 km to 25 km for meteorological parameters [67]. The data is interpolated to 15-minute intervals to enhance temporal granularity, making it suitable for various applications in solar energy forecasting, grid management, and climate analysis. Since irradiance data plays an important role in the generation of PV solar power, the GHI is used in the proposed prediction method.

Incorporating GHI data into the federated model leverages an external variable that directly impacts power output, thereby enhancing feed-in forecasting accuracy. GHI typically exhibits patterns and variations throughout the day and across seasons (see Fig. 4). LSTM and GRU networks are adaptive at learning and exploiting these temporal dynamics, enabling more accurate predictions of feed-in power. As these

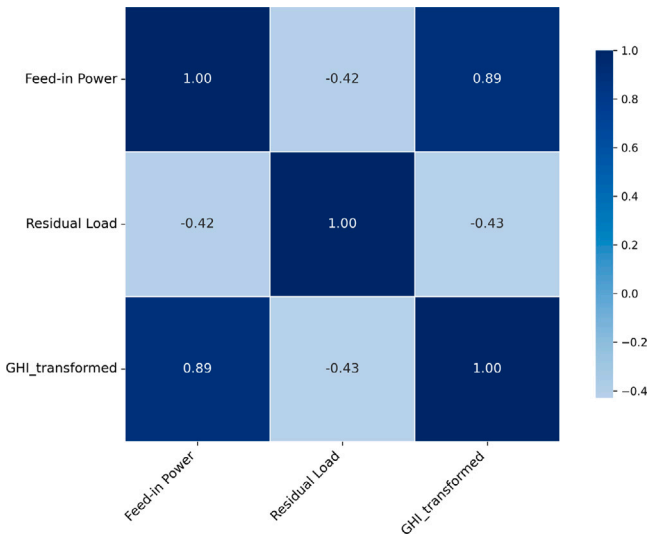

Fig. 5. Linear relationships between feed-in power, residual load and transformed GHI. Each cell shows the Pearson [68] correlation coefficient with values ranging from −1 to 1. There is a strong positive correlation between feed-in and GHI ($r = 0.89$), and a moderate negative correlation between feed-in and load ($r = -0.42$), indicating inverse energy dynamics.

models assimilate new data, they can continuously update their internal representations to account for changes in the relationship between irradiance and feed-in power, thus improving prediction accuracy over time. However, in this real-world scenario, only the feed-in variable requires predictions to enhance low-voltage management.

Correlation matrices between GHI, feed-in power, and residual load, as shown in Fig. 5, demonstrate a high linear correlation among these variables, indicating their suitability for the training with the LSTM and GRU models.

3.2. Data preparation

The raw data from the three smart meter gateways had to be correctly processed and transformed to serve as inputs for the DL models. Given that centralized data collection is not permissible for the federated forecasting method, a distributed processing scheme was employed across all data silos. This was feasible due to the uniformity of their feature space, allowing the application of identical preparation steps. This methodology aligns with the principles of horizontal FL [21, 56].

To address inherent shifts in the measurements of PV data, we adjusted the GHI data by six timestamps. Additionally, to mitigate outliers, a moving average with a window size of five time steps was applied to the data. This window size was empirically determined to strike a balance between over-smoothing and insufficient noise reduction, ensuring a higher data quality for the model inputs. Values below 0.01 kW were set to zero to further clean the dataset. Those low-level noises are not relevant for grid operations by the DSO. In addition, data quality anomalies such as duplicates and erroneous measurements have also been cleansed and removed from the data.

By creating lagged features, temporal dependencies can be better captured, allowing the neural networks to learn and leverage the patterns and correlations that exist between past and current values. For this dataset ten lags were generated, providing a balance between capturing sufficient historical information and maintaining manageable model complexity. The energy dataset therefore comprises a total of 33 features.

Subsequently, the data was normalized using the Min-Max scaling to address the differences in measurement scales between feed-in power (W), residual load (W) and GHI (W/m^2), which would otherwise result in unequal contributions to the model fitting. The Min-Max scaler transforms the data by scaling each feature to the range of $[0, 1]$ [69]. This normalization technique is usually used for improving the model convergence for gradient-based optimization algorithms. The scaling was performed separately on each data silo to adhere to the horizontal FL principle, ensuring that no data was shared across silos during preprocessing.

The average hourly-based values and the data distributions of the three prepared input variables are displayed in Fig. 6. As shown, the highest average grid feed-in from PV generation occurs at midday, when the sun is at its zenith and solar radiation is strongest. In contrast, residual load and thus demand from the low-voltage grid increases substantially more in the evening hours, when the sun goes down. This recurrent pattern is advantageous for training powerful LSTM and GRU models, as they are particularly capable of recognizing such trends in sequential data [37,42].

3.3. Training approach I: Baseline

In the baseline approach, the following two training methods are applied to the LSTM and GRU networks: (1) local, isolated learning per household resp. data silo and (2) centralized learning via a single point of truth (i.e. server).

For the local approach, training is conducted independently on each data silo without sharing data across silos. This method results in a model evaluation being silo-specific. In addition, the test data for the model inference task also remains in the silos and is not shared with other data silos. This training approach does not require moving the data. However, compared to other approaches, the individual data silos cannot benefit from the entirety of the training data of all data silos and are instead completely dependent on the individual silo data. This training approach also does not provide a data privacy guarantee.

In the centralized approach, the data from all silos is transferred to a single location (e.g., a server). The DL models are trained on this centralized dataset and evaluated with centralized test data. This traditional approach promises the highest model performance as all data is directly available and no model bias is generated by Non-IID data. However, data collection and centralization often involves a great deal of effort and is generally not permitted for sensitive data, except for research as in this study, due to data protection regulations. This also applies to the energy dataset used here, which contains confidential meter data from households in the test area.

For initializing the model weights in the DL models, the Xavier [70] initialization is applied in the local and centralized training approach and is defined as following:

$$W \sim \mathcal{U}\left(-\sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}}, \sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}}\right), \tag{2}$$

where n_{in} and n_{out} represent the number of output neurons from the previous layer and output neurons from the current layer, and $\mathcal{U}$ denotes a uniform distribution. This initialization contributes to keeping the scale of the gradients the same in all layers, achieving faster convergence and reducing the risk of vanishing and exploding gradients.

It should be noted that for the reasons stated above, the baseline approaches are only considered as a point of reference in order to gain insights into the relative quality of FL approaches. An overview of the different training approaches followed in this study is also illustrated in Fig. 7.

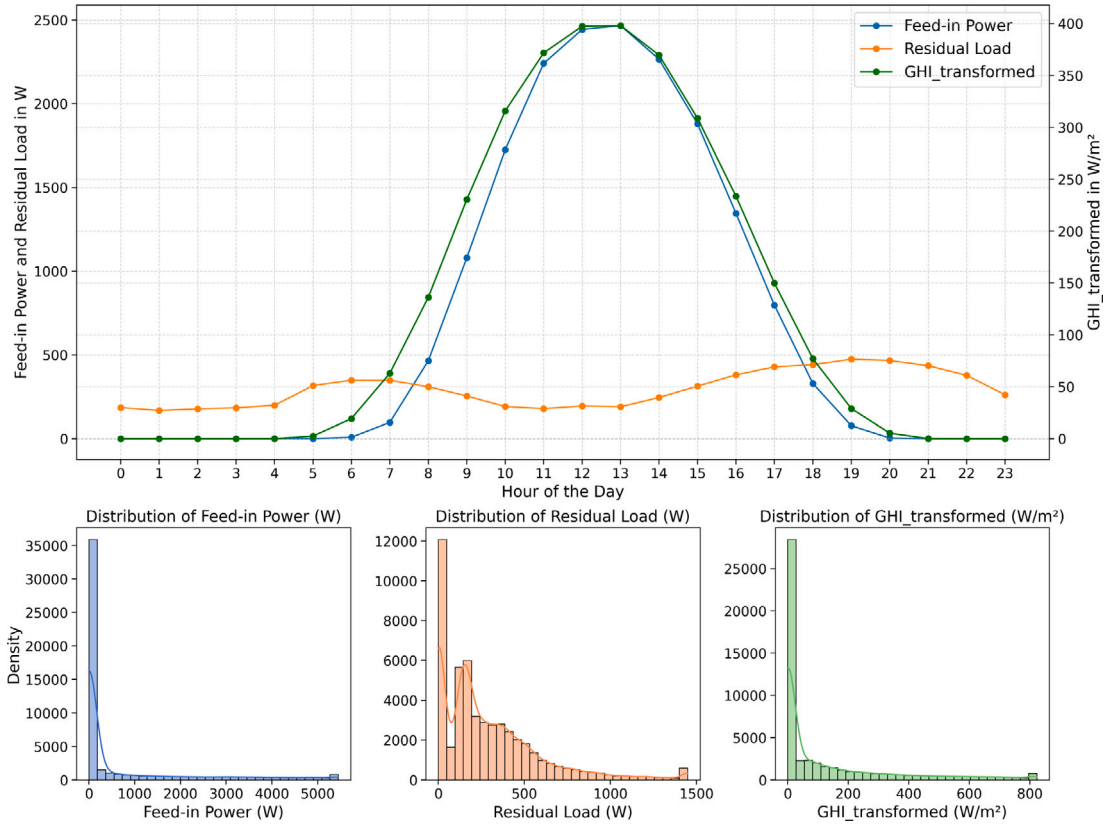

Fig. 6. Average hourly aggregated power metrics after data pre-processing with histograms and kernel density estimates showing the distributions of feed-in power, residual load and the transformed GHI of a single residential household. The zero values indicate periods of inactivity, which are typical in energy datasets due to intermittent generation or residual load (e.g. night time). As the PV systems were connected directly to the power grid, no batteries were used.

	Training Approaches for Enhancing Electrical Feed-in Forecasting			
Configuration Setups	**Baseline**		**Federated**	
	Local	**Centralized**	**FedAvg without DP**	**FedOpt with DP**
Initialization (with k Clients / Data Silos S)	Per Household 	All at one Place  Central Server	Random initialization by FL Server 	Random initialization by FL Server with DP 
Training Procedure (with LSTM and GRU Networks)	Train locally on all **S_1 ... S_k** Silos **for e epochs**	Train on one single server with all data **for e epochs**	Train and aggregate all local models from **S_1 ... S_k** Silos **for e epochs**	Train and aggregate all local models from **S_1 ... S_k** Silos **for e epochs** and employ **DP** into the global model
Fetching Results (with Rolling Cross-Validation (5 Folds))	Evaluate each Silo **S_1 ... S_k** separately and report average **RSME** and **R^2**	Evaluate central model and report average **RSME** and **R^2**	Evaluate global model and report average **RSME** and **R^2**	Evaluate global model and report average **RSME** and **R^2**
Comparing each Learning Approach				

Fig. 7. Overview and procedures of the different training approaches. Each approach is detailed with respect to its initialization with clients/data silos, training procedure utilizing LSTM and GRU networks, and fetching results through rolling cross-validation (CV). The model performance metrics root mean squared error (RMSE) and R^2 are used for evaluating each training approach.

3.4. Training approach II: Federated

In pursuing the federated approach, experiments with FedAvg without DP and experiments with FedAdam and FedYogi with DP are conducted. For those comparison experiments, we group FedAdam and FedYogi together under the term FedOpt to provide a more consistent and concise overview (see Fig. 7).

Using meter data obtained from the smart meter gateways, three different data silos are created for the simulated FL system. Each data silo is a representative of a household installed with a PV system and connected to the electrical grid of the test area (see Fig. 3).

For the model initialization, the Xavier distribution (see Eq. (2)) is here not feasible because it assumes a more balanced and homogeneous data distribution across all data silos. In fact, the Xavier initialization was designed for centralized datasets. Its statistical assumption does not hold for FL and would potentially cause training instability. Moreover, Xavier initialization is based on the assumption of uniform data distribution, which is rarely the case in federated settings. Therefore, for the FL experiments, random initialization is used to create the global model.

3.4.1. Federated optimization

Instead of using a weighted average of the model weight matrices as shown in Eq. (1), the model aggregation strategy can also be updated. As suggested by [24] the FedYogi algorithm can be applied for the model aggregation when the data silos are Non-IID. Formally, Eq. (1) can be changed as follows:

$$\theta_{t+1} = \theta_t - \eta \frac{1}{m} \sum_{i=1}^{m} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \nabla F_i(\theta_t), \tag{3}$$

where θ_t represents the global model parameters at iteration t, η is the learning rate, m is the number of joint data silos in the FL system, $\mathcal{D}_i$ denotes the dataset size of silo i, $|\mathcal{D}|$ is the total dataset size, and $\nabla F_i(\theta_t)$ is the gradient of the loss function with respect to the model parameters θ_t for data silo i.

The FedYogi algorithm introduces adaptive learning rates for each parameter, which supports in dealing with the heterogeneity of the silo data. It leverages an adaptive momentum term, similar to those used in common optimization algorithms such as Adam, to adjust the learning rates based on the history of the gradients [24,26]. This allows the algorithm to converge more efficiently and effectively even in the presence of Non-IID data across different data silos.

The update rules for the FedYogi algorithm can then be described as:

$$v_{t+1} = v_t - (1 - \beta_1) \cdot \nabla F(\theta_t)^2 \cdot \text{sign}(v_t - \nabla F(\theta_t)^2) \tag{4}$$

$$\theta_{t+1} = \theta_t - \eta \frac{1}{\sqrt{v_{t+1} + \epsilon}} \frac{1}{m} \sum_{i=1}^{m} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \nabla F_i(\theta_t), \tag{5}$$

where v_t is an auxiliary variable representing the adaptive second moment estimate at iteration t, β_1 is the momentum term, and ϵ is a small constant (typically $\epsilon = 10^{-8}$) to avoid division by zero. FedYogi ensures that the FL training process is robust to the statistical diversity of the silo data, ensuring a faster and more stable convergence of the global model than standard FedAvg in FL systems with Non-IID data [24,26,27].

FedAdam is a variant of the Adam optimization algorithm tailored for FL [24]. It adapts the learning rates based on the first and second moments (β_1 and β_2) of the gradients, similar to the standard Adam optimizer, but incorporates aggregation and update steps on the coordinator server site from FedAvg [26,71]. Formally, the update rule for FedAdam can be expressed as:

$$g_t = \frac{1}{m} \sum_{i=1}^{m} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \nabla F_i(\theta_t) \tag{6}$$

$$\theta_{t+1} = \theta_t - \eta \frac{\beta_1 m_t + (1 - \beta_1) g_t}{\sqrt{\beta_2 v_t + (1 - \beta_2) g_t^2} + \epsilon}, \tag{7}$$

where g_t is the global gradient of the loss function with respect to the model parameters at time step t, m_t is the first moment vector (mean of the gradients) and v_t is the second moment vector (uncentered variance of the gradients) at time step t, and ϵ is a small constant added for numerical stability. However, it is worth noting that Eq. (7) compresses both the first and second moment updates into the model parameter update.

3.4.2. Differential privacy

The entire data security in FL systems is based on the fact that the training data is not moved in the data silos (data privacy by design). Advanced attack techniques such as model membership inference attack [27,63,64] nevertheless allow the attacker to draw conclusions about the training data. However, by integrating the model update strategy with DP, the data privacy aspect of FL is increased and DL-based attacks are made more difficult.

DP is a privacy-preserving enhancement technique that provides a quantifiable measure of the privacy level of a dataset. This is achieved by introducing a controlled amount of noise when responding to queries on the data [17,20,23]. There is a need to balance the amount of noise added: excessive noise can render computations less useful, while insufficient noise compromises the privacy of the underlying training data. DP introduces the concept of a privacy-loss parameter, denoted by ϵ, which represents the magnitude of noise added for each computation on the data. This *privacy budget* quantifies the information exposed by the computation before the privacy level is considered inadequate. The formula for DP is defined as follows:

$$Pr[M(D) \in Z] \leq e^{\epsilon} Pr\left[M(\bar{D}) \in Z\right] + \delta, \tag{8}$$

where Pr denotes probability, M is the federated model, D and $\bar{D}$ are two neighboring training datasets differing by only one element, and Z is a set of possible outputs. This definition implies that M is ϵ-differentially private. The term δ accounts for a small probability of failure, providing an upper bound on the likelihood that the mechanism deviates from the privacy guarantee.

The advantage of DP is that it offers a measurable guarantee of privacy. However, in practice, determining the appropriate level of privacy and setting the optimal value of ϵ can be challenging [60,63].

Incorporating FedOpt with DP in the proposed FL scheme (see Fig. 7) combines the optimization efficiency of FedYogi and FedAdam with the privacy guarantees of DP, thereby protecting sensitive and confidential data during federated training.

3.4.3. Updated model aggregation strategy

FL with weighted FedAvg without DP is the standard case and often already achieves solid model performances [18,20,22,26,57]. However, weighted FedAvg only has a certain degree of data privacy. Using an advanced federated aggregation strategy such as FedOpt with DP can improve the model generalization capability for Non-IID data silos and ensure a high level of data privacy. Therefore, in this study, a combination of FL with FedOpt (i.e., FedAdam and FedYogi) with DP is used in the proposed forecasting method. The complete model update strategy for FL training with FedYogi and DP is also described in pseudo code in Algorithm 1.

Algorithm 1 Federated training process with FedYogi and DP

Input: $r_{\max}$: max. round number, C: All data silos in FL system, S: FL Server, n: number of randomly selected clients per FL round, E: epochs, J: error, L_{adj}: adjusted loss, θ: neural network weights, V: gradient, α: learning rate, θ_0: saved model weights, τ: FedYogi parameter, η: learning rate, β_1, β_2: hyperparameters for Yogi update, ϵ: privacy budget, δ: probability of failing to achieve ϵ-differential privacy, σ: noise scale, C: clipping norm
S.init() // Initialize coordinator server S
for each client c in C **do**
  c.load_data()
  c.init_model(θ_0)
  c.connect(S)
end for
while round $r < r_{\max}$ **do**
  Select n random clients C_n
  for each client c in C_n **do**
    $\theta_r^c \leftarrow \theta_r^S$ // Receive server weights
    for epoch e in E **do**
      $X, Y = c$.data.get_batch(e)
      $\hat{Y} = c$.model.predict(X)
      $J(\theta_r^c) = L_{\text{adj}}(Y, \hat{Y})$ // Loss function
      $g_r^c = \nabla J(\theta_r^c)$ // Compute gradient
      $g_r^c = g_r^c / \max(1, \|g_r^c\|/C)$ // Clip the gradient
      $v_r^c = \beta_2 \cdot v_{r-1}^c + (1-\beta_2) \cdot (g_r^c)^2$ // Update variance
      $\theta_{r+1}^c = \theta_r^c - \eta \cdot g_r^c / (\sqrt{v_r^c} + \tau)$ // Yogi update step
    end for
    $g_r^c = g_r^c + \mathcal{N}(0, \sigma^2 \cdot I)$ // Add Gaussian noise for DP
    $S \leftarrow \theta_{r+1}^c$ // Send weights to server
  end for
  $\theta_{r+1}^S = \sum_{c=0}^{n} \frac{n_c}{n} \theta_{r+1}^c$ // Weighted averaging of n data silos
  $r = r + 1$
end while

4. Experimental settings

In this section, the evaluation methodology, loss functions, and model hyperparameters used in the comparative experiments are elucidated.

For the conducted experiments described in Section 5, an FL system with three data silos was simulated on a local GPU server. As elaborated in Section 3.4, a data silo represents a household with a PV system from the test area. The coordinator server for the model aggregation step was also hosted on the local server. Docker containers were employed to maintain data separation between the data silos and the FL framework used in this study was integrate.ai [72]. In summary, the test system consisted of the following components:

- One GPU server with 2 RTX 3090 GPUs, 64 GB RAM, and i9-9900K CPU @ 3.60 GHz.
- Three Docker containers, one for each data silo.
- One Docker container for the coordinator server.

4.1. Model architectures definition

The LSTM and GRU model architectures were each defined with two layers: a single LSTM or GRU layer followed by an multi-layer perceptron (MLP). These relatively shallow DL models offer a more robust and less complex federated training, so LSTMs and GRUs with more layers were not considered in this study. In all training approaches, the CUDA accelerator was used. The model configuration for LSTM and GRU is displayed in Table 1.

For all LSTM and GRU models, rectified linear unit (ReLU) activation functions were used between the layers. These ReLU activation functions introduce non-linearity to the models, enabling the learning of complex patterns in the data by allowing only positive values to pass through and setting negative values to zero, which also reduces the vanishing gradient risk [73].

Table 1
Configuration of LSTM and GRU model architectures.

Layer type	Input size	Output size
LSTM Input Layer (for LSTM only)	128	64
GRU Input Layer (for GRU only)	128	64
MLP Output Layer (for both)	64	1

4.2. Evaluation approach

To evaluate the model performance of the baseline and federated models, a rolling k-fold CV approach was used. This method involves dividing the time series data into k-folds, where each fold is used as a test set while the remaining k-1 folds are used for training in a sequential order [74,75]. The rolling technique ensures that the temporal order of the data is preserved, which is necessary in time series forecasting tasks. For evaluating the experiments a rolling 5-fold CV is applied on each data silo for each configuration. A graphical illustration of this is shown in Fig. 8.

The model performance metrics RMSE and R^2 were used to assess the accuracy and explanatory power of the prediction models (see Fig. 7). RMSE provides a measure of the average magnitude of the errors between the predicted and actual values, with lower values indicating better model performance. On the other hand, the R^2 metric is a statistical measure used to evaluate the quality of the fit of a model [76]. It indicates the proportion of the variance in the dependent variable that is predictable from the independent variables, with values closer to 1 suggesting a higher quality of fit. The metric is formally defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{9}$$

where the numerator is the sum of the squared differences between the observed values (y_i) and the predicted values ($\hat{y}_i$). The denominator of Eq. (9) measures observed values (y_i) and the mean of the observed values ($\bar{y}$). Both performance metrics are commonly used in downstream tasks of ML-based time series forecasting [12,13,31,32,76].

After each experiment, the model performance of the federated models was compared with the centralized base models to determine the effectiveness of the proposed prediction method. Therefore, the results from the rolling 5-fold CV were aggregated to provide a comprehensive evaluation of model performance across different data splits, ensuring the robustness and reliability of the experimental findings.

4.3. Loss function

The RMSE and R^2 performance metrics are used to compare each training method (see Fig. 7). However, these performance metrics are not used for the internal validation process during model training. Given that the energy prediction task exhibits characteristics of a regression problem, the mean squared error (MSE) has been used as the loss function. The MSE is defined as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{10}$$

where y_i represents the actual values, $\hat{y}_i$ represents the predicted values, and n is the number of observations. The MSE measures the average of the squares of the errors, providing a quadratic penalty for large errors and thus emphasizing discrepancies between the predicted and actual values [77]. For the global model, a weighted average loss of MSE (see Eq. (10)) is calculated based on the size of the data silos.

It might be worth to note that loss functions in neural networks are differentiable functions used to quantify the accuracy of predictions.

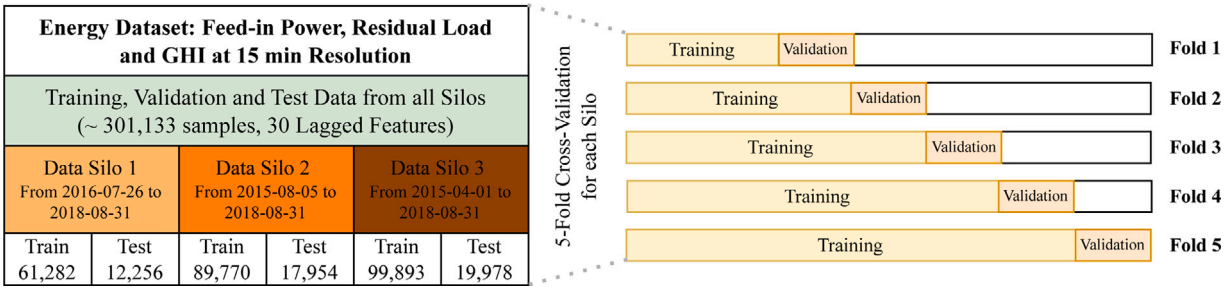

Fig. 8. Data splitting of the energy dataset with rolling 5-fold CV within each data silo (i.e. households with meter data). Each silo contains training data from different starting points, resulting in Non-IID siloed data.

Different loss functions can lead to different optimal points on the loss surface. However, other loss functions, such as quantile loss or custom loss functions, were not pertinent to the objectives of this paper and, therefore, were not considered further.

4.4. Hyperparameter search

The correct choice of hyperparameters and their values can positively influence the model generalization capability. Therefore, the GridSearchCV [78,79] method was used in this study to select suitable hyperparameters for the training of the LSTM and GRU models. The procedure was as follows:

1. A grid of relevant hyperparameters and their possible values was defined. This grid was applied equally to both model architectures (LSTM and GRU).
2. GridSearchCV uses CV to evaluate the models. The dataset was split into $k = 5$ folds and trained similarly to the fold-wise approach depicted in Fig. 8.
3. A model was trained and validated for each combination of hyperparameters in the grid.
4. After evaluating all hyperparameter combinations, the configuration with the best average MSE value on the validation folds was selected.

Other hyperparameter search methods, such as HyperBand Search [80] and hyperparameter optimization frameworks such as Optuna [81], can also lead to optimal hyperparameter values. However, the goal of this paper is to improve data privacy in smart grid analytics and compare federated LSTM with GRU models (see Fig. 7). Therefore, GridSearchCV has been considered as a valid approach for determining appropriate hyperparameter values.

An overview of the functions and optimized hyperparameters used in the comparison experiments is summarized in Table 2.

The maximum gradient norm is used to clip the gradients during the backpropagation process in neural networks [82]. It prevents the gradients from exploding, which is also a typical issue in RNN based model trainings [38]. The maximum value for gradient clipping has been identified using GridSearchCV so that the gradients will be scaled down to ensure their norm is at most 4 (see Table 2).

To prevent overfitting during model training early stopping as a DL regularization technique was used in all training approaches. The training was stopped if the validation loss did not improve for a specified number of epochs (see Table 2).

The different learning rates from Table 2 were chosen based on the observation during grid testing that a too low learning rate for the federated training leads to a reduced global model performance. For example, the study's simulations showed that at a learning rate of $\eta = 0.001$, the global model tended to get stuck in a suboptimal local minimum, resulting in subpar performance. This issue was observed for both the LSTM and GRU architectures, as the low learning rate hindered the model's ability to make effective progress towards the global minimum. Thus, we conclude that identical hyperparameters cannot be directly transferred from centralized learning to FL problems due to the differing loss surfaces. Therefore, hyperparameters need to be separately adjusted for both training approaches.

Table 2
Summary of the selected functions and hyperparameters for the experiments. Hyperparameters were determined using GridSearchCV.

Functions and Hyperparameter	Value
Nr. of Model Parameters (LSTM)	67,813
Nr. of Model Parameters (GRU)	54,405
Nr. of Lag Variables	30
Nr. of K-Folds	5
Batch Size B	128
Epochs (Local Training only) e_{max}	120
Federated Training Rounds r_{max}	120
Loss Criterion	MSE
Performance Metrics	RMSE, R^2
SGD Optimizer with Momentum μ	0.4
Learning Rate η_{local}	0.001
Learning Rate $\eta_{federated}$	0.01
Early Stopping Regularization E_{stop}	50
Dropout Rate	0.2
Differential Privacy-budget ϵ	6
Maximum Gradient Norm	4
Federated Aggregation Functions	FedAvg, FedOpt

5. Experimental results

The results of the experiments performed in this study, as depicted in Fig. 7, are presented and discussed in this section. Unless otherwise described, each experiment was conducted and evaluated according to the settings defined in Section 4.

5.1. Model comparisons

All LSTM and GRU models used in this study were not pre-trained and were instead trained from scratch based on the energy dataset with rolling 5-fold CV, where feed-in power was the target variable in all experiments. The GHI and the residual load as well as the lagged variables formed the feature set (see Fig. 8).

The maximum number of training rounds, $r_{max} = 120$ and $e_{max} = 1$ were used for the federated approaches. Thus, a local model in a data silo was fully trained after one epoch. Then a federated aggregation phase was started and the local model updates were aggregated on the server-side with a FL algorithm, as shown and described in Fig. 3. This reduced the training time at silo level and led to a more stable global model more quickly. As a side effect, this model configuration also optimizes the execution of the model training on low-computational thin or edge clients, as there is no permanent hardware load as with centralized or local only training with higher number of epochs.

Since early stopping patience criteria [83] were used in the experiments, the number of epochs trained on each fold deviated from the

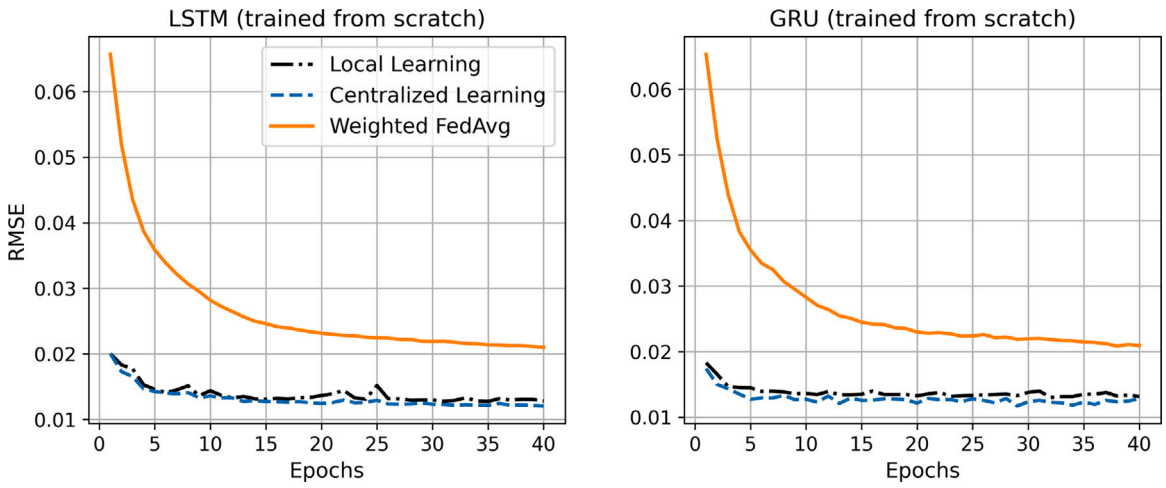

Fig. 9. Average rolling 5-fold CV RMSE on the energy test data with three data silos using different training approaches (local training, centralized training, and weighted FedAvg).

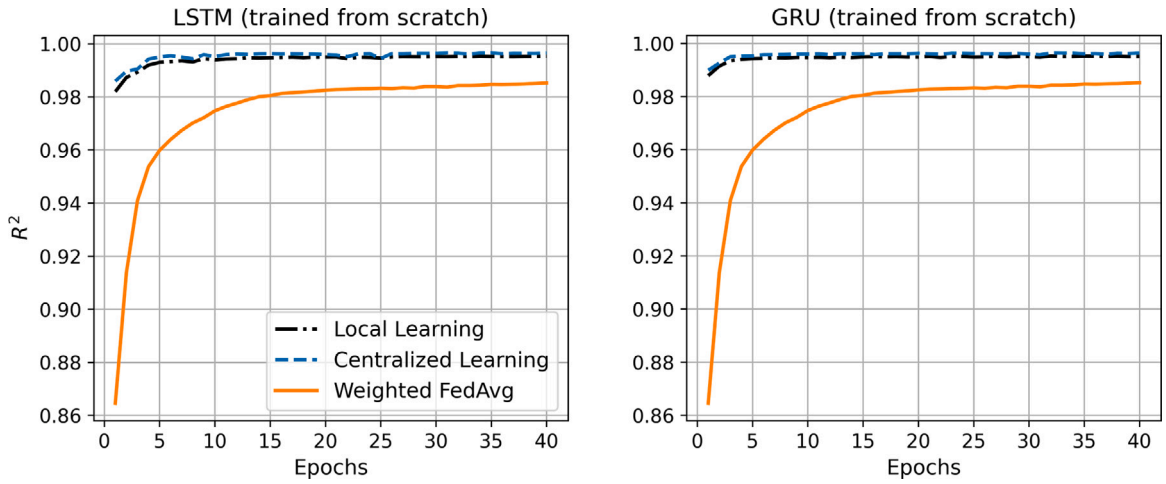

Fig. 10. Average rolling 5-fold CV R^2 on the energy test data with three data silos using different training approaches (local training, centralized training, and weighted FedAvg).

originally set value (see Table 2). However, all experiments conducted showed that there were no relevant changes in the prediction accuracy after 40 training rounds, so we capped the performance metrics in the comparison plots to 40 rounds.

5.1.1. Baseline approach

In the local training approach, LSTM and GRU models were independently trained from scratch and evaluated within each data silo. The testing results (i.e. RMSE and R^2) from each data silo were then aggregated by averaging and grouped as local learning (see Fig. 7). Based on the selected model configurations, the local LSTM and GRU models demonstrated high performance in all three data silos already after a few epochs. As shown in Fig. 9, the centralized trained LSTM model performed best with an RMSE value of 0.0121 after 40 epochs. In contrast, for the local training, the LSTM and GRU models performed slightly worse with RMSE values of 0.0128 and 0.0131, respectively. The models in the third data silo, which had the highest number of training and test data, showed similar results in the experiments. It cannot be entirely ruled out that the smaller number of test data points in the first data silo (7722 fewer timestamps than in the third silo) led to slight overfitting, even if rolling CV was used. However, the performance results of the other local models on the downstream task, as well as their corresponding training accuracy (MSE), were also at similar levels. The reasons for the generally high model performance observed in all experiments were:

- The energy dataset possesses high data quality due to tailored data preparation and normalization steps. Outliers and erroneous sensor measurements in the PV systems or meter gateways were less than 0.5%,
- a low number of features to avoid the curse of dimensionality effect [84], and
- optimized hyperparameter were used.

However, the local training approach, as mentioned in Section 3.3, is not an option for the DSO, as constant access to each data silo (i.e. household) must be guaranteed.

For the centralized approach, the training and test data from the individual data silos were consolidated on a central GPU server. An arithmetic mean was applied to overlapping time series features. As depicted in Fig. 10, the LSTM and GRU models with centralized learning performed similarly to the local models in the respective data silos. Both model architectures achieved R^2 values of 0.9948 (LSTM) and 0.9953 (GRU) in fewer than 10 epochs, indicating a high generalization capability. It is worth noting that in this baseline training approach, more training data did not necessarily lead to a much better model performance compared to the locally trained models. This was because the training data for the centralized learning setting largely consisted of the averaged data from the three individual data silos (intersection), so that the model performances shown for the centralized baseline approach from Fig. 9 and Fig. 10 were to be expected.

Table 3
Model performances of different data-driven forecasting methods on the test data. The values displayed are the mean values with standard deviation across the epochs resp. training rounds. Best values for each metric and model architecture are in bold.

Training Approach	RMSE	R^2	RMSE	R^2
	LSTM from scratch		GRU from scratch	
Local Learning	0.0138 ± 0.0015	0.9940 ± 0.0025	0.0137 ± 0.0009	0.9946 ± 0.0011
Centralized Learning	$\mathbf{0.0132 \pm 0.0016}$	$\mathbf{0.9954 \pm 0.0021}$	$\mathbf{0.0128 \pm 0.0009}$	$\mathbf{0.9958 \pm 0.0013}$
FL with FedAvg ($\epsilon = 6$)	0.0332 ± 0.0091	0.9677 ± 0.0223	0.0332 ± 0.0089	0.9677 ± 0.0223
FL with FedAdam ($\epsilon = 6$)	0.0350 ± 0.0079	0.9677 ± 0.0019	0.0327 ± 0.0100	0.9697 ± 0.0193
FL with FedYogi ($\epsilon = 6$)	0.0345 ± 0.0075	0.9712 ± 0.0187	0.0342 ± 0.0076	0.9768 ± 0.0187

Table 4
Model training times of the experiments using CUDA acceleration. Each experiment was repeated 10 times to assess variability. The shortest time for each model architecture is in bold.

Training Approach	LSTM from scratch	GRU from scratch
Local Learning	**6 m 37 s ± 7 s**	**2 m 32 s ± 4 s**
Centralized Learning	6 m 58 s ± 5 s	2 m 33 s ± 6 s
FL with FedAvg ($\epsilon = 0$)	12 m 15 s ± 10 s	8 m 25 s ± 8 s
FL with FedAdam ($\epsilon = 0$)	11 m 50 s ± 9 s	7 m 45 s ± 6 s
FL with FedYogi ($\epsilon = 0$)	11 m 35 s ± 12 s	7 m 32 s ± 9 s
FL with FedAvg ($\epsilon = 6$)	9 m 10 s ± 8 s	5 m 50 s ± 5 s
FL with FedAdam ($\epsilon = 6$)	8 m 55 s ± 7 s	5 m 40 s ± 4 s
FL with FedYogi ($\epsilon = 6$)	8 m 45 s ± 6 s	5 m 35 s ± 5 s

5.1.2. Federated approach

Compared to the baseline models, the federated LSTM and the GRU with weighted FedAvg performed slightly worse on both metrics and required more training rounds to converge (see Fig. 9). The training behavior of the federated LSTM and GRU was more uniform compared to the centrally and locally trained models because weighted FedAvg tends to smooth out fluctuations as it averages the model updates from multiple data silos [56]. Each data silo contained a different amount of training data and the averaging step resulted in a more stable and consistent model update after each federated round. This behavior is also noticeable when compared to local models trained on heavily Non-IID data, which can exhibit higher variability [17,25]. Fig. 9 and Fig. 10 also show that the performance of the global model approaches the baseline models as the number of federated training rounds in the network increases. Although a FL model can achieve high acceptable prediction accuracies, it is worth noting that FL models cannot outperform a centralized model when using the same dataset, due to the way the model weights are aggregated.

Regarding the model architectures, the LSTM and GRU models achieved similar prediction accuracy across all comparative experiments. Although the GRU models performed slightly better than the LSTM models, the federated GRU models showed slightly more oscillation, but provided a faster training. The comparable performance values can be attributed to the fact that both architectures are types of RNNs designed to capture long-term dependencies in the data (see Fig. 2).

The RMSE and R^2 performance metrics of the conducted experiments are summarized in Table 3.

For the training times, Table 4 indicates that the GRU models consistently completed training faster than the LSTM models, primarily due to the lighter model architecture of GRUs compared to LSTMs (see Fig. 2). However, a lower privacy budget ϵ resulted in longer federated training times compared to higher privacy budgets. This suggests that the DP process of adding noise to the model weight matrices slightly slows down FL. Nevertheless, the model architecture remains the most relevant factor affecting computational time.

It is worth to note that FL incurs additional computational overhead, such as communication bandwidth costs between the coordinator server and data silos, as well as the overhead from the DP mechanism. This explains why the local and centralized training approaches are generally faster. However, the actual training times can also vary depending on the optimization landscape, which is mainly influenced by the dataset.

5.2. Impact of differential privacy

To federate the models with DP, we used the PyTorch-based Opacus framework [85], which is also a component of integrate.ai [72], and tested different levels of data privacy to investigate the impact of added privacy on the generalizability of the federated models.

The selection of an appropriate privacy budget ϵ impacts the performance of federated models and is challenging to determine. A higher privacy budget results in less noise added to the global model during the aggregation step. For highly sensitive data, such as medical patient data, a very low privacy budget (often below 1) is recommended [23, 63]. However, an excessively low privacy budget can degrade the model performance, causing issues such as exploding or vanishing gradients due to the excessive artificial noise added to the gradients.

Concerning the training times from Table 4, if the global model meets the convergence criteria, the training time is more likely to increase with a smaller privacy budget (lower ϵ). However, this is not always the case, as it also depends on the absolute value of ϵ. For example, a reduction in ϵ from 80 to 60 can have a smaller impact on the training time than a reduction from 10 to 8. As ϵ becomes smaller, the noise added becomes more disruptive to the training process, thus increasing training time more greatly.

It is essential to strike a balance between sufficient data privacy and effective model training. The study's experiments indicate that for federating the energy dataset with a privacy budget of 6, data privacy is slightly reduced compared to lower budgets, but the global model's performance remains largely stable and high. Fig. 11 and Fig. 12 visualizes the comparisons of individual federated LSTM and GRU models with different FL algorithms and DP across different privacy budgets. However, the average drop in federated model performance across training rounds is only 0.0200 RMSE for FedAvg with DP compared to centralized learning

The DP parameters, such as the maximum value for gradient clipping in conjunction with the chosen privacy budget, may also influence the federated model's performance. The clipping function works as a sensitivity factor in DP and specifies the maximum amount that the model output can change [86]. Nevertheless, there is no optimal method currently available to determine the best combination of these parameters. Furthermore, the mutual influence of these parameters on each other is scarcely addressed in the existing literature, highlighting a valuable area for further research.

5.3. Impact of data distributions

As illustrated in Fig. 8, the time series in the three data silos had different start times, resulting in a Non-IID simulated FL system. The extended FL algorithms, FedAdam (see Eq. (7)) and FedYogi (see Eq. (3)), effectively handled the Non-IID siloed energy data. Figs. 11 and 12 show that for LSTM and GRU models, FedAdam and FedYogi achieved a better model performance more quickly than weighted FedAvg within the first five rounds of training. However, beyond this initial period, there were no significant differences between weighted FedAvg, FedAdam, and FedYogi, indicating that the influence of Non-IID data is reduced when the data variation between the data silos is

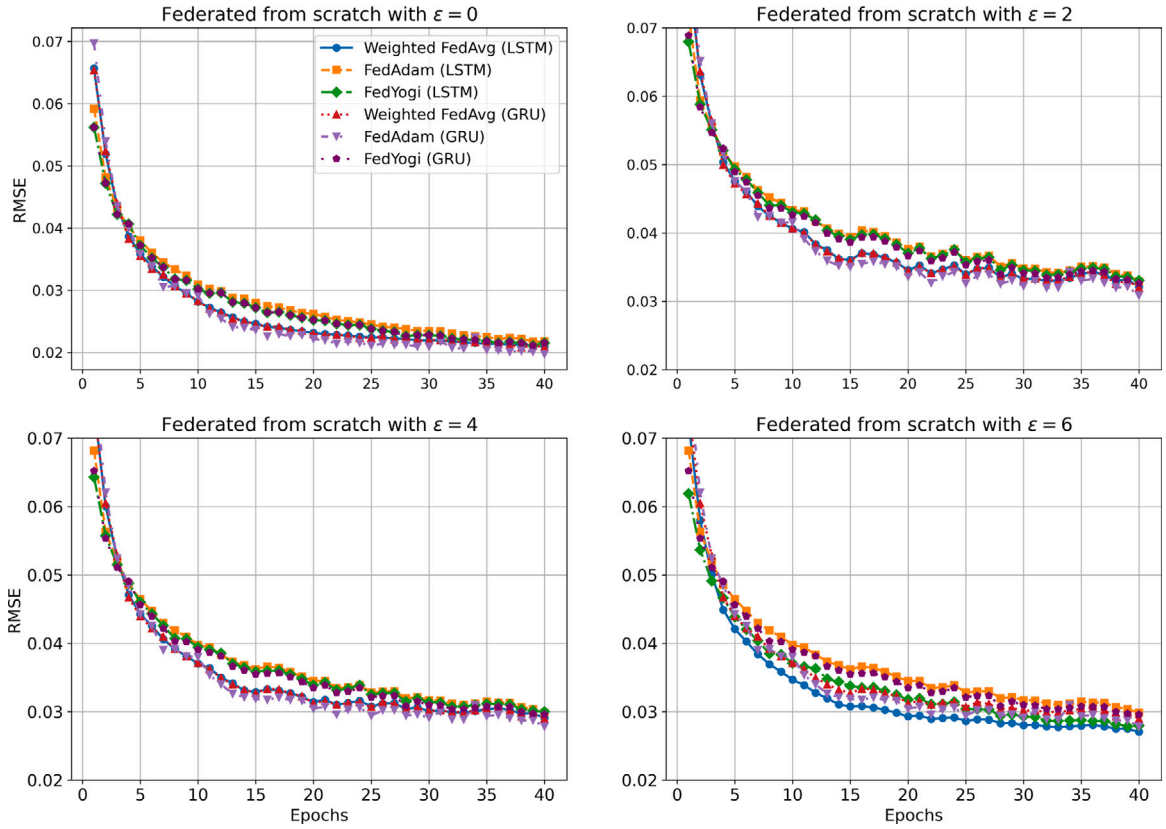

Fig. 11. Average rolling 5-fold CV RMSE comparing FedAvg, FedAdam and FedYogi for federated LSTM and GRU models with different DP budget ϵ.

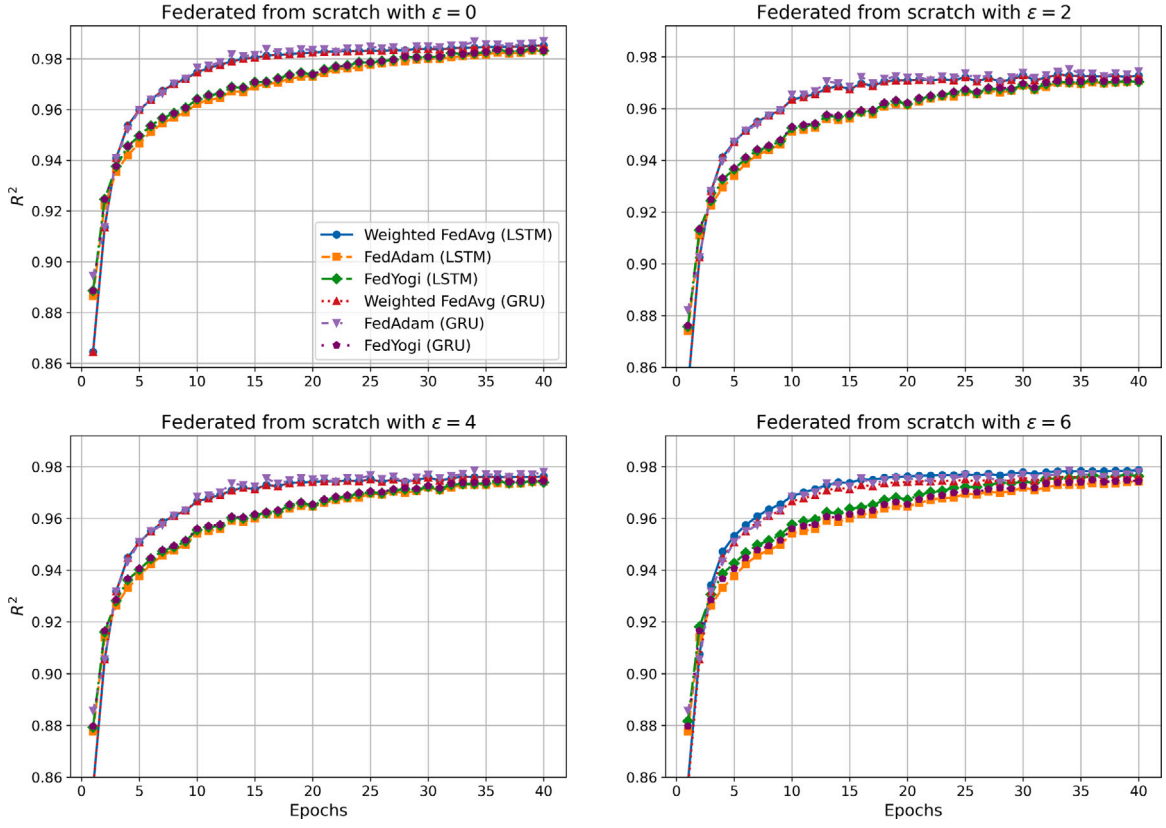

Fig. 12. Average rolling 5-fold CV R^2 comparing FedAvg, FedAdam and FedYogi for federated LSTM and GRU models with different DP budget ϵ.

minimal. Overall, all three FL algorithms successfully reduced the variance and achieved stable model convergence in less than 40 training rounds.

It is also worth noting, that weighted FedAvg may exhibit slower initial convergence compared to locally and centralized trained models (see Fig. 10), especially when the data across silos is Non-IID. Nevertheless, advanced FL algorithms have superior generalization capabilities, which enhances its robustness against overfitting to any specific data silo distribution.

Another reason why the impact of Non-IID data played a less significant role in this study, compared to DP, was the number of data silos. In an FL system, the probability of encountering heavily Non-IID silos increases with the number of joined data silos, which can hinder the global model's convergence.

5.4. Implications for smart grid operator

For the local training approach, the training data remains within the data silos. Despite achieving high prediction accuracy for feed-in power, this training method has several inherent drawbacks:

- Model quality dependence: The quality of the model depends solely on the respective silo's data.
- Data quality issues: A data silo with highly imbalanced, underpopulated, or erroneous data results in a poorer prediction model that cannot be compensated for by other data silos with higher data quality. Collaborative DL is not possible with this approach, as the model weights remain local.
- Scalability challenges: Scaling with n households in the test area is complex and cost-intensive due to the lack of a centralized access.

The situation is similar with centralized learning. Access to the data silos and the transfer of sensitive training and test data is required, which is often not feasible in real-world scenarios.

Although Table 3 indicates that the prediction accuracy of federated models with DP and a privacy budget of $\epsilon = 6$ is generally somewhat worse than local and centralized models without DP, the FL approach with DP offers distinct advantages. The privacy by design principle of FL ensures that silo data remains local, but only the combination of FL with DP guarantees a high level of data privacy during the exchange of raw model weights in the federated training process, thereby reducing the risk of DL based attacks such as model membership inference.

By using FedOpt algorithms (FedAdam or FedYogi), federated recurrent models such as LSTM and GRU can be effectively trained on heavily Non-IID silo data, facilitating scalability with additional data silos. The experiments also demonstrated that GRU models are trained faster and reached a stable performance plateau more quickly than LSTM models. However, as shown in Fig. 11, the absolute performance differences between GRU and LSTM models are relatively small.

This study recommends the proposed federated training approach utilizing GRU, FedOpt, and the inclusion of DP with a moderate privacy budget to the DSO for forecasting feed-in power in the low-voltage grid. The introduced prediction method adheres to state-of-the-art data protection regulations and eliminates the need for direct access to silo data (i.e., meter gateways) in households for smart grid analytics. Furthermore, various experiments in this study have demonstrated that the performance of federated models is only marginally inferior to that of locally or centrally trained models.

5.5. Summary of results

The results and the main contributions in this study can be summarized as follows:

- Our work is distinguished as a first comparison study that employs FL to train and compare LSTM and GRU models in a privacy-preserving manner for feed-in power prediction in low-voltage grids.
- The dataset, unique in its composition, encompasses data from authentic residential PV systems located in South Germany with a 15-minute time resolution, augmented with regional solar irradiance information, presenting a real-world scenario for the application of FL.
- With the implementation of DP, an improvement in the data privacy of federated LSTM and GRU models is introduced to meet privacy regulations in smart grids.
- This study proposes a federated-driven forecasting approach that is characterized by its accuracy, computational efficiency and focus on maximizing data privacy. Furthermore, the severity of Non-IID problems between data silos is reduced by introducing an advanced FL algorithm, resulting in an improved and stable federated model.

5.6. Future directions for federated learning research

In future work, we plan to extend the proposed prediction method to encompass additional real-world households within the test area. This extension will allow us to observe the behavior of the training process as more households join the FL system, providing insights into scalability and robustness. It will be particularly interesting to investigate the impact of the Non-IID scenario when data silos with adverse data distributions or low data quality are included. Understanding how these factors influence model performance and generalizability is relevant for practical FL applications.

Moreover, investigating the application of our prediction method to other domains beyond the energy sector could provide valuable insights into its generalizability and adaptability. Future studies might explore the integration of additional privacy-preserving techniques, such as secure multi-party computation [58], homomorphic encryption [60] or model personalization [14], to further enhance data security in FL systems.

Additionally, further research is required in the area of federated hyperparameter tuning with DP and other optimization parameters such as gradient clipping. This involves exploring the potential of automated methods to optimize federated hyperparameters while maintaining privacy guarantees. We think there is substantial untapped potential in this area, which could lead to significant improvements in security of FL and training efficiency.

In summary, our future work aims to address scalability, robustness, and optimization in FL, with a focus on maintaining strong privacy guarantees. These efforts will contribute to advancing the field and ensuring the practical applicability of FL in various real-world scenarios.

6. Conclusion

This study introduced a novel data privacy-preserving feed-in power forecasting method with federated learning (FL) and differential privacy (DP) for distributed system operators (DSOs) operating in low-voltage grids. To achieve this, experiments were conducted using three years of real-world meter data from a test area in a southern German city. This meter data included the feed-in power resulting from a surplus of photovoltaic (PV) power, as well as the residual load from the low-voltage grid when the PV system's generation was insufficient. The data was enriched with meteorological information such as global horizontal irradiance. Three households with PV systems from the test area were selected for the study object. In the context of FL, each household represented an isolated data silo, with no interaction between the silos. Consistent data preparation across the data silos improved the data quality.

An experimental comparison was conducted between federated training approaches, local silo-based training, and centralized learning, where all energy data was consolidated in a single location. For the

model architectures, the recurrent deep learning models long short-term memory (LSTM) and gated recurrent unit (GRU) were trained and federated from scratch, as they are well-suited for capturing long-term dependencies and temporal structures in time series data.

The experimental results demonstrated that the choice of the privacy budget for DP affects the performance of the federated LSTM and GRU models. A low privacy budget resulted in inferior model performance compared to a higher privacy budget, which, in turn, led to a lower data privacy. For the feed-in power forecasting, we considered a moderate privacy budget of $\epsilon = 6$ to be an appropriate trade-off between the model inference quality and inevitable data privacy.

The unequal data distribution among the data silos, due to different starting times, did not substantially affect the generalization ability of the federated models when using weighted FedAvg, FedAdam, or FedYogi. Only during the first initial federated training rounds FedAdam and FedYogi achieved a stable plateau in the loss surface faster than weighted FedAvg. Regardless of the chosen federated aggregation strategy, this study showed that there were larger deviations between federated LSTM and GRU models in the early training rounds, with GRU models converging faster than LSTM models. The evaluation was conducted using a rolling 5-fold cross-validation approach to detect potential overfitting. Compared to local silo-wise learning and centralized learning, the federated approaches performed slightly worse. However, federated GRU models with FedYogi and DP achieved a sufficiently high prediction accuracy, with an RMSE of 0.0342 and an R^2 of 97.68%. In contrast to other data-driven training approaches, the proposed federated forecasting method complies with common data protection standards and enables the DSO to efficiently perform smart grid analyses without the need for extensive centralization of meter data.

CRediT authorship contribution statement

Pascal Riedel: Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis. **Kaouther Belkilani:** Writing – review & editing, Validation, Project administration, Data curation. **Manfred Reichert:** Writing – review & editing, Validation, Supervision. **Gerd Heilscher:** Writing – review & editing, Supervision, Resources, Project administration. **Reinhold von Schwerin:** Writing – review & editing, Validation, Supervision, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This project is supported by the project InterBDL (Project funding indicator 01MV23025A) and Project OrPHEus (Project No. 608930). The data preparation process was further supported by David Gögelein, a research associate of the Technical University of Applied Sciences at Ulm.

Data availability

The data that has been used is confidential.

References

[1] Zhu Hangyu, Xu Jinjin, Liu Shiqing, Jin Yaochu. Federated learning on non-IID data: A survey. Neurocomputing 2021;465:371–90. http://dx.doi.org/10.1016/j.neucom.2021.07.098, URL https://www.sciencedirect.com/science/article/pii/S0925231221013254.

[2] Ruf H. Limitations for the feed-in power of residential photovoltaic systems in Germany – an overview of the regulatory framework. Sol Energy 2018;159:588–600. http://dx.doi.org/10.1016/J.SOLENER.2017.10.072.

[3] Bayer Benjamin, Matschoss P, Thomas Heiko, Marian A. The german experience with integrating photovoltaic systems into the low-voltage grids. Renew Energy 2018;119:129–41. http://dx.doi.org/10.1016/J.RENENE.2017.11.045.

[4] Weniger J, Tjaden T, Bergner J, Quaschning V. Sizing of battery converters for residential PV storage systems. Energy Procedia 2016;99:3–10. http://dx.doi.org/10.1016/J.EGYPRO.2016.10.092.

[5] Goodrich A, James T, Woodhouse M. Residential, commercial, and utility-scale photovoltaic (PV) system prices in the United States: Current drivers and cost-reduction opportunities. Technical report, Golden, Colorado: National Renewable Energy Laboratory; 2012.

[6] Johanning Simon, Abitz Daniel, Scheller Fabian, Bruckner Thomas. The influence of financial benefits and peer effects on the adoption of residential rooftop photovoltaic systems. In: 2023 19th international conference on the European energy market. EEM, 2023, p. 1–16. http://dx.doi.org/10.1109/EEM58374.2023.10161765.

[7] Fraunhofer Institute. Energy charts: Energy mix in Germany 2023. 2023, https://www.energy-charts.info/charts/energy_pie/chart.htm?l=de&c=DE&interval=year&year=2023, Accessed: 2023-04-10.

[8] Chen Shuo, Heilscher Gerd, Ebe Falko, Kondzialka Christoph, Idlbi Basem. Analyse der integration von PV-systemen in smart grids. 2021.

[9] Shafiullah M, Ahmed Shakir D, Al-Sulaiman F. Grid integration challenges and solution strategies for solar PV systems: A review. IEEE Access 2022;10:52233–57. http://dx.doi.org/10.1109/ACCESS.2022.3174555.

[10] Quakernack Lars, Kelker M, Haubrock J. Deep reinforcement learning for autonomous control of low voltage grids with focus on grid stability in future power grids. In: 2022 IEEE PES innovative smart grid technologies conference Europe (ISGT-Europe). 2022, p. 1–5. http://dx.doi.org/10.1109/ISGT-Europe54678.2022.9960416.

[11] Meliani Meryem, Barkany AE, Abbassi IE, Darcherif A, Mahmoudi M. Energy management in the smart grid: State-of-the-art and future trends. Int J Eng Bus Manag 2021;13. http://dx.doi.org/10.1177/18479790211032920.

[12] Pereira Sara, Canhoto Paulo, Salgado Rui. Development and assessment of artificial neural network models for direct normal solar irradiance forecasting using operational numerical weather prediction data. Energy and AI 2024;15:100314. http://dx.doi.org/10.1016/j.egyai.2023.100314, URL https://www.sciencedirect.com/science/article/pii/S2666546823000861.

[13] González-Peña David, García-Ruiz Ignacio, Díez-Mediavilla Montserrat, Dieste-Velasco Mª Isabel, Alonso-Tristán Cristina. Photovoltaic prediction software: Evaluation with real data from northern Spain. Appl Sci 2021;11(11). http://dx.doi.org/10.3390/app11115025, URL https://www.mdpi.com/2076-3417/11/11/5025.

[14] Widmer Fabian, Nowak Severin, Bowler Benjamin, Huber Patrick, Papaemmanouil Antonios. Data-driven comparison of federated learning and model personalization for electric load forecasting. Energy and AI 2023;14:100253. http://dx.doi.org/10.1016/j.egyai.2023.100253, URL https://www.sciencedirect.com/science/article/pii/S2666546823000253.

[15] Real António Corte, Luz G Pontes, Sousa JMC, Brito MC, Vieira SM. Optimization of a photovoltaic-battery system using deep reinforcement learning and load forecasting. Energy and AI 2024;16:100347. http://dx.doi.org/10.1016/j.egyai.2024.100347, URL https://www.sciencedirect.com/science/article/pii/S2666546824000132.

[16] Brauneck Alissa, Schmalhorst Louisa, Kazemi Majdabadi Mohammad Mahdi, Bakhtiari Mohammad, Völker Uwe, Saak Christina Caroline, et al. Federated machine learning in data-protection-compliant research. Nat Mach Intell 2023;5(1):2–4. http://dx.doi.org/10.1038/s42256-022-00601-5.

[17] Riedel Pascal, von Schwerin Reinhold, Schaudt Daniel, Hafner Alexander, Späte Christian. ResNetFed: Federated deep learning architecture for privacy-preserving pneumonia detection from COVID-19 chest radiographs. J Healthc Inform Res 2023;7(2):203–24. http://dx.doi.org/10.1007/s41666-023-00132-7.

[18] McMahan H Brendan, Moore Eider, Ramage Daniel, Hampson Seth, Arcas Blaise Agüera y. Communication-efficient learning of deep networks from decentralized data. 2016, http://dx.doi.org/10.48550/ARXIV.1602.05629, URL https://arxiv.org/abs/1602.05629.

[19] McMahan H Brendan, Ramage Daniel, Talwar Kunal, Zhang Li. Learning differencially private recurrent language models. In: International conference on learning representations. 2018.

[20] Zhang Xiaojin, Kang Yan, Chen Kai, Fan Lixin, Yang Qiang. Trading off privacy, utility, and efficiency in federated learning. ACM Trans Intell Syst Technol 2023;14:98:1–89:31. http://dx.doi.org/10.1145/3595185.

[21] Zeng Dun, Liang Siqi, Hu Xiangjing, Wang Hui, Xu Zenglin. FedLab: A flexible federated learning framework. J Mach Learn Res 2023;24:1–7.

[22] Xing Huanlai, Xiao Zhiwen, Qu Rong, Zhu Zonghai, Zhao Bowen. An efficient federated distillation learning system for multi-task time series classification. IEEE Trans Instrum Meas 2022;71:1–12. http://dx.doi.org/10.1109/TIM.2022.3201203.

[23] Dwork C. Differential privacy. 2006, p. 1–12. http://dx.doi.org/10.1007/11787006_1, This paper introduces differential privacy, a concept that captures the increased risk to one's privacy incurred by participating in a database, proposing a framework that offers strong privacy guarantees while allowing for the extraction of useful statistics.

[24] Reddi Sashank, Charles Zachary, Zaheer Manzil, Garrett Zachary, Rush Keith, Konečný Jakub, et al. Adaptive federated optimization. 2020, http://dx.doi.org/10.48550/ARXIV.2003.00295, URL https://arxiv.org/abs/2003.00295.

[25] Hsu Tzu-Ming Harry, Qi Hang, Brown Matthew. Measuring the effects of non-identical data distribution for federated visual classification. 2019, http://dx.doi.org/10.48550/ARXIV.1909.06335, URL https://arxiv.org/abs/1909.06335.

[26] Riedel Pascal, Reichert Manfred, Von Schwerin Reinhold, Hafner Alexander, Schaudt Daniel, Singh Gaurav. Performance analysis of federated learning algorithms for multilingual protest news detection using pre-trained DistilBERT and BERT. IEEE Access 2023;11:134009–22. http://dx.doi.org/10.1109/ACCESS.2023.3334910.

[27] Zhang Geer, Zhu Songyang, Bai Xiaoqing. Federated learning-based multi-energy load forecasting method using CNN-attention-LSTM model. Sustainability 2022. http://dx.doi.org/10.3390/su141912843.

[28] Rajagukguk Rial A, Ramadhan Raden AA, jin Lee Hyun. A review on deep learning models for forecasting time series data of solar irradiance and photovoltaic power. Energies 2020. http://dx.doi.org/10.3390/en13246623, A Review on Deep Learning Models for Forecasting Time Series Data of Solar Irradiance and Photovoltaic Power. This review discusses deep learning models for solar irradiance and PV power prediction, highlighting LSTM, GRU, and their hybrid models' performance and applicability.

[29] Chen Biaowei, Lin P, Lin Yaohai, Lai Y, Cheng Shuying, Chen Zhicong, Wu Lijun. Hour-ahead photovoltaic power forecast using a hybrid GRA-LSTM model based on multivariate meteorological factors and historical power datasets. IOP Conf Ser: Earth Environ Sci 2020;431. http://dx.doi.org/10.1088/1755-1315/431/1/012059, Hour-ahead photovoltaic power forecast using a hybrid GRA-LSTM model based on multivariate meteorological factors and historical power datasets. This study presents a grey relational analysis (GRA) combined with a long short-term memory recurrent neural network (LSTM RNN) model for short-term forecasting of PV power plants, showing robust performance in prediction..

[30] AlKandari Mariam, Ahmad I. Solar power generation forecasting using ensemble approach based on deep learning and statistical methods. Appl Comput Inform 2020. http://dx.doi.org/10.1016/j.aci.2019.11.002, Solar power generation forecasting using ensemble approach based on deep learning and statistical methods. This research proposes a hybrid model combining machine learning methods with Theta statistical method, including LSTM and GRU models, for accurate prediction of solar power generation..

[31] Gong Guanhua, Lou Ke, Yin Jie, Li Dongyv. Forecast of photovoltaic power generation based on GRU. In: Proceedings of the 2022 6th international conference on electronic information technology and computer engineering. 2022, http://dx.doi.org/10.1145/3573428.3573477, Forecast of photovoltaic Power Generation Based on GRU. This paper introduces a model using a gated recurrent unit (GRU) for short-term power prediction of photovoltaic plants, demonstrating high accuracy and stability compared to CNN and LSTM models..

[32] Xue J, Hu Xucheng, Chen Haifeng, Zhou Gang. Research on LSTM-xgboost integrated model of photovoltaic power forecasting system. In: 2022 14th international conference on intelligent human-machine systems and cybernetics (IHMSC). 2022, p. 22–5. http://dx.doi.org/10.1109/ihmsc55436.2022.00014, Research on LSTM-XGBoost Integrated Model of Photovoltaic Power Forecasting System. This study analyzes the combination of Long Short-Time Memory (LSTM) algorithm and Extreme Gradient Boosting (XGBoost) for photovoltaic forecasting, showing higher forecasting accuracy compared to popular models including GRU.

[33] Kumar D, Mathur HD, Bhanot S, Bansal R. Forecasting of solar and wind power using LSTM RNN for load frequency control in isolated microgrid. Int J Modelling Simul 2020;41:311–23. http://dx.doi.org/10.1080/02286203.2020.1767840.

[34] Gangopadhyay Tryambak, De Somnath, Liu Qisai, Mukhopadhyay Achintya, Sen Swarnendu, Sarkar Soumik. An LSTM-based approach to detect transition to lean blowout in swirl-stabilized dump combustion systems. Energy and AI 2024;16:100334. http://dx.doi.org/10.1016/j.egyai.2023.100334, URL https://www.sciencedirect.com/science/article/pii/S2666546823001064.

[35] Li Zheng, Luo Xiaorui, Liu Mengjie, Cao Xin, Du Shenhui, Sun Hexu. Short-term prediction of the power of a new wind turbine based on IAO-LSTM. Energy Rep 2022;8:9025–37. http://dx.doi.org/10.1016/j.egyr.2022.07.030.

[36] De Soham, Smith Samuel L, Fernando Anushan, Botev Aleksandar, Cristian-Muraru George, Gu Albert, et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. 2024, arXiv:2402.19427.

[37] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9:1735–80. http://dx.doi.org/10.1162/neco.1997.9.8.1735, This paper introduces LSTM, a novel, efficient, gradient-based method that can learn to bridge minimal time lags in excess of 1000 discrete-time steps by enforcing constant error flow through constant error carousels within special units.

[38] Hu Yuhuang, Huber Adrian EG, Anumula Jithendar, Liu Shih-Chii. Overcoming the vanishing gradient problem in plain recurrent networks. 2018, ArXiv arXiv:1801.06105, The Recurrent Identity Network (RIN) overcomes the vanishing gradient problem in plain recurrent networks, achieving competitive performance and faster convergence compared to IRNNs and LSTMs in various sequence learning tasks.

[39] Yu Yong, Si Xiaosheng, Hu Changhua, Zhang Jianxun. A review of recurrent neural networks: Lstm cells and network architectures. Neural Comput 2019;31(7):1235–70. http://dx.doi.org/10.1162/neco_a_01199.

[40] Jailani Nur Liyana Mohd, Dhanasegaran Jeeva Kumaran, Alkawsi G, Alkahtani A, Phing Chen Chai, Baashar Yahia, et al. Investigating the power of LSTM-based models in solar energy forecasting. Processes 2023. http://dx.doi.org/10.3390/pr11051382, LSTM models, both independent and hybrid, effectively forecast solar energy output using time-series data, outperforming other conventional machine learning methods.

[41] Skrobek Dorian, Krzywanski Jaroslaw, Sosnowski Marcin, Kulakowska Anna, Zylka Anna, Grabowska Karolina, et al. Prediction of sorption processes using the deep learning methods (long short-term memory). Energies 2020;13(24). http://dx.doi.org/10.3390/en13246601, URL https://www.mdpi.com/1996-1073/13/24/6601.

[42] Cho Kyunghyun, van Merrienboer Bart, Gulcehre Caglar, Bahdanau Dzmitry, Bougares Fethi, Schwenk Holger, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014, arXiv:1406.1078, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). This paper introduces the GRU, designed to adaptively capture dependencies of different time scales.

[43] Kisvari Adam, Lin Zi, Liu Xiaolei. Wind power forecasting – a data-driven method along with gated recurrent neural network. Renew Energy 2021;163:1895–909. http://dx.doi.org/10.1016/j.renene.2020.10.119, The novel data-driven approach using gated recurrent neural networks improves wind power forecasting accuracy and reduces computational costs compared to Long Short-term Memory (LSTM) algorithms.

[44] Mahjoub S, Chrifi-Alaoui L, Marhic B, Delahoche L. Predicting energy consumption using LSTM, multi-layer GRU and drop-GRU neural networks. Sensors (Basel, Switzerland) 2022;22. http://dx.doi.org/10.3390/s22114062, The LSTM neural network provides better short-term power consumption forecasting with fewer prediction errors and finer precision than the GRU and Drop-GRU methods, enabling advanced load shedding decisions..

[45] Skrobek Dorian, Krzywanski Jaroslaw, Sosnowski Marcin, Kulakowska Anna, Zylka Anna, Grabowska Karolina, et al. Implementation of deep learning methods in prediction of adsorption processes. Adv Eng Softw 2022;173:103190. http://dx.doi.org/10.1016/j.advengsoft.2022.103190, URL https://www.sciencedirect.com/science/article/pii/S0965997822000977.

[46] Li Hao, Zhou Qi, Tian Jing, Lin Xiaoyu. Energy demand forecasting for an office building based on random forests. In: 2020 IEEE 4th conference on energy internet and energy system integration (EI2). 2020, p. 29–32. http://dx.doi.org/10.1109/ei250167.2020.9347021, This study applied random forests to forecast the energy demand of office buildings, demonstrating the feasibility and accuracy of this method in managing energy consumption without prior historical load data..

[47] Talekar Bhushan. A detailed review on decision tree and random forest. Biosci Biotechnol Res Commun 2020. http://dx.doi.org/10.21786/BBRC/13.14/57, This paper discusses the limitations and advantages of decision trees and random forests, highlighting their popularity in various fields due to their ability to handle classification and regression tasks effectively..

[48] Jin Bingchu, Hu Zesheng, Quan Mingrui, Wang Yuxin, Wang Jianhua, Huang He. Short-term power load forecasting based on self-adaptation random forest. In: 2021 IEEE 5th conference on energy internet and energy system integration (EI2). 2021, p. 3121–6. http://dx.doi.org/10.1109/EI252483.2021.9713339, presents a Random Forest-based method for short-term power load forecasting, which includes a regression tree parameter optimization algorithm to enhance forecasting accuracy. The model was shown to perform better than traditional machine learning prediction models, offering a simpler and more efficient forecasting solution.

[49] Cattani Gilles. Combining data envelopment analysis and random forest for selecting optimal locations of solar PV plants. Energy and AI 2023;11:100222. http://dx.doi.org/10.1016/j.egyai.2022.100222, URL https://www.sciencedirect.com/science/article/pii/S2666546822000684.

[50] Pallonetto Fabiano, Jin Changhong, Mangina Eleni. Forecast electricity demand in commercial building with machine learning models to enable demand response programs. Energy and AI 2022;7:100121. http://dx.doi.org/10.1016/j.egyai.2021.100121, URL https://www.sciencedirect.com/science/article/pii/S2666546821000690.

[51] Fischer Thomas G, Krauss C. Deep learning with long short-term memory networks for financial market predictions. European J Oper Res 2017;270:654–69. http://dx.doi.org/10.1016/j.ejor.2017.11.054.

[52] Gasparin Alberto, Lukovic Slobodan, Alippi Cesare. Deep learning for time series forecasting: The electric load case. 2019, arXiv:1907.09207.

[53] Zhang Lefeng, Zhu Tianqing, Xiong P, Zhou Wanlei, Yu P. A robust game-theoretical federated learning framework with joint differential privacy. IEEE Trans Knowl Data Eng 2023;35:3333–46. http://dx.doi.org/10.1109/TKDE.2021.3140131.

[54] Wen Jie, Zhang Zhixia, Lan Yang, Cui Zhihua, Cai Jianghui, Zhang Wensheng. A survey on federated learning: challenges and applications. Int J Mach Learn Cybern 2023;14:513–35. http://dx.doi.org/10.1007/s13042-022-01647-y.

[55] Tang Xinyu, Guo Cheng, Choo Kim-Kwang Raymond, Liu Yining. An efficient and dynamic privacy-preserving federated learning system for edge computing. IEEE Trans Inf Forensics Secur 2024;19:207–20. http://dx.doi.org/10.1109/TIFS.2023.3320611, This paper proposes a dynamic federated edge learning (FEL) scheme that defends against malicious edge servers and devices, improving security and model performance in federated learning systems.

[56] McMahan H Brendan, Moore Eider, Ramage Daniel, Hampson Seth, y Arcas Blaise Agüera. Communication-efficient learning of deep networks from decentralized data. J Mach Learn Res 2017;54:1273–82.

[57] Nguyen Anh, Do Tuong, Tran Minh, Nguyen Binh X, Duong Chien, Phan Tu, et al. Deep federated learning for autonomous driving. In: 2022 IEEE intelligent vehicles symposium. IV, 2022, p. 1824–30. http://dx.doi.org/10.1109/IV51971.2022.9827020.

[58] Wu Changti, Zhang Lei, Xu Lin, Choo Kim-Kwang Raymond, Zhong Liangyu. Privacy-preserving serverless federated learning scheme for internet of things. IEEE Internet Things J 2024;1. http://dx.doi.org/10.1109/JIOT.2024.3380597.

[59] Zhou Ian, Tofigh Farzad, Piccardi Massimo, Abolhasan Mehran, Franklin Daniel, Lipman Justin. Secure multi-party computation for machine learning: A survey. IEEE Access 2024;12:53881–99. http://dx.doi.org/10.1109/ACCESS.2024.3388992.

[60] Zhu Huafei. On the relationship between (secure) multi-party computation and (secure) federated learning. 2020, ArXiv arXiv:2008.02609, Federate learning (FL) is a subset of multi-party computation (MPC) and can be privately computed using various techniques like homomorphic encryption, secure multi-party computation (SMPC), and differential privacy (DP).

[61] Hussien N, Hussien N, Salman Saba Abdulbaqi, Aljanabi Mohammad. Secure federated learning with a homomorphic encryption model. Int J Papier Adv Sci Rev 2023. http://dx.doi.org/10.47667/ijpasr.v4i3.235.

[62] Gupta Shreya, Arora Ginni. Use of homomorphic encryption with GPS in location privacy. In: 2019 4th international conference on information systems and computer networks (ISCON). 2019, p. 42–5. http://dx.doi.org/10.1109/ISCON47742.2019.9036149, This paper discusses the inefficiencies and high computational demands of homomorphic encryption, highlighting its impracticality for real-time applications Gupt.

[63] Ouadrhiri Ahmed El, Abdelhadi Ahmed M. Differential privacy for deep and federated learning: A survey. IEEE Access 2022;10:22359–80. http://dx.doi.org/10.1109/ACCESS.2022.3151670, Differential privacy (DP) effectively protects users' privacy in deep and federated learning by adding noise to datasets or learning parameters, ensuring strong privacy protection in data analysis..

[64] Nasr Milad, Shokri Reza, Houmansadr Amir. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In: 2019 IEEE symposium on security and privacy. SP, 2019, p. 739–53. http://dx.doi.org/10.1109/SP.2019.00065.

[65] Wang Jianyu, Liu Qinghua, Liang Hao, Joshi Gauri, Poor H Vincent. Tackling the objective inconsistency problem in heterogeneous federated optimization. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, editors. Advances in neural information processing systems. Vol. 33, Curran Associates, Inc.; 2020, p. 7611–23, URL https://proceedings.neurips.cc/paper_files/paper/2020/file/564127c03caab942e503ee6f810f54fd-Paper.pdf.

[66] Karimireddy Sai Praneeth, Kale Satyen, Mohri Mehryar, Reddi Sashank, Stich Sebastian, Suresh Ananda Theertha. SCAFFOLD: Stochastic controlled averaging for federated learning. In: III Hal Daumé, Singh Aarti, editors. Proceedings of the 37th international conference on machine learning. Proceedings of machine learning research, Vol. 119, PMLR; 2020, p. 5132–43, URL https://proceedings.mlr.press/v119/karimireddy20a.html.

[67] Cebecauer T, Suri M. Typical meteorological year data: Solargis approach. Energy Procedia 2015;69:1958–69.

[68] Jebli Imane, Belouadha Fatima-Zahra, Kabbaj Mohammed Issam, Tilioua Amine. Prediction of solar energy guided by pearson correlation using machine learning. Energy 2021;224:120109. http://dx.doi.org/10.1016/j.energy.2021.120109, URL https://www.sciencedirect.com/science/article/pii/S0360544221003583.

[69] Fessler JA, Sutton BP. Nonuniform fast Fourier transforms using min-max interpolation. IEEE Trans Signal Process 2003;51(2):560–74. http://dx.doi.org/10.1109/TSP.2002.807005.

[70] Glorot Xavier, Bengio Yoshua. Understanding the difficulty of training deep feedforward neural networks. In: Teh Yee Whye, Titterington Mike, editors. Proceedings of the thirteenth international conference on artificial intelligence and statistics. Proceedings of machine learning research, Vol. 9, Chia Laguna Resort, Sardinia, Italy: PMLR; 2010, p. 249–56, URL https://proceedings.mlr.press/v9/glorot10a.html.

[71] Deng Xiumei, Li Jun, Wei Kang, Shi Long, Xiong Zeihui, Ding Ming, et al. Towards communication-efficient federated learning via sparse and aligned adaptive optimization. 2024, arXiv:2405.17932.

[72] Integrateai. Integrate.ai. 2024, https://www.integrate.ai/, Accessed: 2024-06-18.

[73] Bai Yuhan. RELU-function and derived function review. SHS Web Conf 2022;144:02006. http://dx.doi.org/10.1051/shsconf/202214402006.

[74] Wong Tzu-Tsung, Yeh Po-Yang. Reliable accuracy estimates from k-fold cross validation. IEEE Trans Knowl Data Eng 2020;32(8):1586–94. http://dx.doi.org/10.1109/TKDE.2019.2912815.

[75] scikit-learn developers. TimeSeriesSplit. 2024, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html, Accessed: 2024-06-18.

[76] Colin Cameron A, Windmeijer Frank AG. An R-squared measure of goodness of fit for some common nonlinear regression models. J Econometrics 1997;77(2):329–42. http://dx.doi.org/10.1016/S0304-4076(96)01818-0, URL https://www.sciencedirect.com/science/article/pii/S0304407696018180.

[77] PyTorch Developers. MSELoss. 2024, https://pytorch.org/docs/stable/generated/torch.nn.MSELoss.html, Accessed: 2024-06-18.

[78] Ahmad Ghulab Nabi, Fatima Hira, Ullah Shafi, Salah Saidi Abdelaziz, Imdadullah. Efficient medical diagnosis of human heart diseases using machine learning techniques with and without GridSearchCV. IEEE Access 2022;10:80151–73. http://dx.doi.org/10.1109/ACCESS.2022.3165792.

[79] Scikit-learn. GridSearchCV. 2024, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html, Accessed: 2024-06-17.

[80] Li Lisha, Jamieson Kevin, DeSalvo Giulia, Rostamizadeh Afshin, Talwalkar Ameet. Hyperband: A novel bandit-based approach to hyperparameter optimization. 2018, arXiv:1603.06560.

[81] Akiba Takuya, Sano Shotaro, Yanase Toshihiko, Ohta Takeru, Koyama Masanori. Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. KDD '19, New York, NY, USA: Association for Computing Machinery; 2019, p. 2623–31. http://dx.doi.org/10.1145/3292500.3330701.

[82] Clip Gradient Norm, https://pytorch.org/docs/stable/generated/torch.nn.utils.clip_grad_norm_.html, Accessed: 2024-06-17.

[83] Zhou Wangchunshu, Xu Canwen, Ge Tao, McAuley Julian, Xu Ke, Wei Furu. Bert loses patience: Fast and robust inference with early exit. Adv Neural Inf Process Syst 2020;33:18330–41.

[84] Bach Francis. Breaking the curse of dimensionality with convex neural networks. J Mach Learn Res 2017;18(19):1–53, URL http://jmlr.org/papers/v18/14-546.html.

[85] Yousefpour Ashkan, Shilov Igor, Sablayrolles Alexandre, Testuggine Davide, Prasad Karthik, Malek Mani, et al. Opacus: User-friendly differential privacy library in PyTorch. 2021, arXiv preprint arXiv:2109.12298.

[86] Zhang Xinwei, Chen Xiangyi, Hong Mingyi, Wu Zhiwei Steven, Yi Jinfeng. Understanding clipping for federated learning: Convergence and client-level differential privacy. In: International conference on machine learning, ICML 2022. 2022.