Privacy-Preserving Probabilistic Voltage Forecasting in Local Energy Communities

Jean-François Toubeau, *Member, IEEE,* Fei Teng, *Member, IEEE,* Thomas Morstyn, *Member, IEEE,* Leandro Von Krannichfeldt, and Yi Wang, *Member, IEEE*

***Abstract*—This paper presents a new privacy-preserving framework for the short-term (multi-horizon) probabilistic forecasting of nodal voltages in local energy communities. This task is indeed becoming increasingly important for cost-effectively managing network constraints in the context of the massive integration of distributed energy resources. However, traditional forecasting tasks are carried out centrally, by gathering raw data of end-users in a single database that exposes their private information. To avoid such privacy issues, this work relies on a distributed learning scheme, known as federated learning wherein individuals' data are kept decentralized. The learning procedure is then augmented with differential privacy, which offers formal guarantees that the trained model cannot be reversed-engineered to infer sensitive local information. Moreover, the problem is framed using cross-series learning, which allows to smoothly integrate any new client joining the community (i.e., cold-start forecasting) without being plagued by data scarcity. Outcomes show that the proposed approach achieves improved performance compared to non-collaborative (locally trained) models, and is able to reach a trade-off between privacy and performance for different architectures of deep learning networks.**

***Index Terms*—Differential privacy, Deep learning, Federated learning, Heterogeneous data, Voltage forecasting.**

I. Introduction

THE deployment of distributed energy resources, such as photovoltaic (PV) generation, electric vehicles and heat pumps, leads to higher energy flows in distribution systems, which are thus increasingly subject to stressed operating conditions [1]. To comply with security constraints while avoiding costly infrastructure investments, one solution is to proactively manage local energy exchanges [2]. This can be achieved through local energy communities (LECs) which gather end-users into organized entities, wherein energy resources are pooled and allocated to reach common (e.g., economical, or environmental) objectives [3]. However, to ensure optimal coordination between resources, LECs need to be informed with accurate predictions of the future system state [4].

The work is supported via the energy transition funds project 'Adabel' organized by the FPS economy, S.M.E.s, Self-employed and Energy. (*Corresponding author: Yi Wang.)*

J.-F. Toubeau is a research fellow of the National Fund of Scientific Research (FNRS) in the Power Systems and Markets Research Group, University of Mons, Belgium.

F. Teng is with the Department of Electrical and Electronic Engineering, Imperial College London, London, UK.

T. Morstyn is with the School of Engineering Science at the University of Edinburgh, Edinburgh EH9 3JW, United Kingdom. (e-mail: Thomas.Morstyn@ed.ac.uk)

L. Von Krannichfeldt is with the Power Systems Laboratory, ETH Zurich, 8092 Zurich, Switzerland. (leandro.von.krannichfeldt@hotmail.com)

Y. Wang is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China. (yiwang@eee.hku.hk).

Our objective is thus to develop a new framework for the short-term probabilistic forecasting of nodal voltage magnitudes in LECs, exploiting the information from smart metering devices. The voltage information is indeed essential to support an optimized operation of the system, e.g., by providing insights on how much flexibility (such as curtailment of renewable generation, and load shifting) needs to be gathered from the local end-users to prevent voltage violations [5].

In contrast with state estimation that is used for identifying current voltage values in support of real-time grid management [6], [7], state forecasting is used for estimating future voltage values to enable a pro-active network scheduling (to avoid costly and suboptimal redispatch actions).

Although load and renewable energy forecasting tasks have been thoroughly studied [8], the literature on voltage forecasting is still very sparse. In [9]–[11], nodal generation and consumption are firstly predicted, and then embedded into a network model to calculate nodal voltages. However, these methods rely on the perfect knowledge of the network parameters, which are usually uncertain (e.g., the phases on which each client is connected are often unknown). To bypass this limitation, data-driven approaches have been developed. The voltage levels are represented by vector autoregressive (VAR) processes in [12]. To avoid such linear models, a deep learning model is proposed in [13], where it is shown that high accuracy can be achieved by monitoring only a few strategic buses. In [14], an ensemble approach (combining different regression models) is introduced for the deterministic voltage prediction. This work has been extended in [15] and [16] in a probabilistic setting.

All these algorithms assume that private data can be freely accessed from a centralized location. In a competitive environment, the nodal (e.g., smart meter) data are owned by end-users, who may be reluctant to share this information (as it may reveal private aspects such as home occupancy, routines, and usage of specific appliances). To address privacy concerns, an efficient solution is to rely on distributed learning. In [17], the alternating direction method of multipliers (ADMM) has been used in the context of renewable energy forecasting to train a ridge linear quantile regression model. Since ADMM does not guarantee privacy (as adversaries can recover the data if they have access to the intermediate calculations [18]), the procedure is enriched with data encryption in [19] for training autoregressive models. Alternatively to such ADMM-based techniques (which are limited to convex models [20]), federated learning (FL) has recently emerged [21] to build complex (non-linear) forecasters, such as tree-based [22] or

deep learning [23] models. To that end, FL relies on a distributed setting where each client locally computes an update to the model (based on its own data). A central server then aggregates all client-side updates to compute a new global model, such that no raw data are communicated.

However, although FL complicates data inference since no centralized server holds all the information, it is not sufficient to ensure data privacy. Indeed, it has been shown that trained models may be reverse-engineered to extract detailed input information from the end-users involved in the training phase [24], [25]. It is thus essential to provide guarantees that the trained forecasting model protects the privacy of individual databases [26]. Cryptography-based methods have been proposed to hedge confidentiality breaches, but the high computational burden of current schemes is a barrier to their real-life applicability and integration in an energy-efficient society [27]. Indeed, encrypting data not only significantly increases their size (in the order of 100x–10,000x), but also lead to higher computation times (in the order of 100x) [28].

In this paper, we therefore augment FL (which lacks rigorous privacy guarantees against inference attacks) with differential privacy (DP). The principle of DP is to inject noise into the training procedure, which is calibrated in such a way that the privacy leakage of any sensitive information can be bounded and quantified [29]. In particular, the methodology is tailored to achieve user-level privacy, i.e., enforcing that the dataset of any client has a limited impact on the learned model, thus preventing inference of local raw information.

Another important aspect for the LEC is to smoothly accommodate end-users with different histories, such as households with newly installed smart meter. Hence, traditional learning strategies wherein local features (e.g., past load and PV generation) from all nodes are aggregated into the same input vector should be avoided. Indeed, the sites with few historical measures inherently limit the number of samples to train the model (i.e., a longer history of other end-users cannot be used). In this paper, we therefore develop a cross-learning approach [30], which mitigates the problem of (local) data scarcity by treating each end-user as a different sample to train a single, generic model. By generalizing to all samples from all clients of the community, the model endogenously learns common patterns from neighboring nodes (thus capturing space dependencies). Moreover, this transfer of learning enables cold-start forecasting for end-users with no historical data. Overall, the contributions of the paper are three-fold.

1) We leverage distributed learning (to avoid data exchanges in the training), differential privacy (to prevent inference of private data from the trained model) and cross-series learning (to accommodate end-users with diverse measurements histories) in an innovative framework dedicated to the private probabilistic forecasting of voltage levels in LECs. The approach is developed for different deep learning models, due to their ability to capture complex high-dimensional dependencies.
2) We bridge the gap between data utilization and protection by keeping track of the privacy loss accumulated over the learning course. This is achieved by ensuring that the training procedure satisfies Rényi differential privacy, which offers a tight analysis of privacy consumed. To that end, we formulate the model training as a sampled Gaussian mechanism, which consists in drawing a random subset of end-users at each training stage followed by the addition of Gaussian noise.
3) We provide insights for optimally tuning different architectures of deep learning models trained with user-level differential privacy. In addition, we explore the trade-off between improving model performance while maintaining privacy of raw data. We show how this trade-off can be adapted based on the LEC privacy preferences, which is key to enhance the engagement and satisfaction of all end-users.

Outcomes show the added value of the proposed privacy-preserving framework in comparison with fully private forecasting (where each end-user trains its own model with its own data), thus highlighting the interest for end-users to collaboratively train a joint community-wide model. Overall, the generic nature of the method paves the way towards the integration of privacy-enhancing techniques in smart grids.

In the rest of the paper, the different building blocks to construct the private voltage forecaster are introduced in section II. These concepts are combined and enriched in section III to propose a new probabilistic model with provable privacy guarantees. The section IV defines the different models used as a benchmark. These models are then tested on a radial low-voltage network, and the resulting outcomes are analyzed in section V. Finally, the main conclusions are summarized in section VI.

II. Problem formulation and background

In this section, we firstly formulate the probabilistic voltage forecasting problem using cross-series learning. Then, we introduce the underlying deep learning-based model. Finally, we elaborate on the concepts of federated learning and differential privacy, which will be used (in section III) to enrich the forecasting model with strong privacy guarantees.

A. Model description

The objective is to generate privacy-preserving probabilistic forecasts of nodal voltages within low-voltage LECs. Nodal voltages are governed by intricate time correlations as well as space dependencies arising from network constraints (i.e., neighboring buses are likely to exhibit similar voltage patterns). Capturing such space-time dependencies is a challenging task since the size of the LEC may evolve (e.g., with new homes) and the LEC may thus be composed of end-users with different histories, including some with very few measurements.

Hence, as shown in the case study (section V-A), the traditional solution of jointly predicting all nodal voltages in a single instance of the forecasting model may face data scarcity (and thus poor performance) since the total number of training samples is limited by the end-user with the smallest database (so that many relevant data are lost). Moreover, such an approach is not scalable since increasing the number of

TABLE I
INPUT FEATURES OF THE NODAL VOLTAGE FORECASTER.

Past data $\mathbf{x}_{:t_0,n}^{(p)}$	past (local) load, past (aggregated) PV generation, past (aggregated) imports-exports, calendar information (weekday, hour of the day)
Known future data $\mathbf{x}_{t_1:T,n}^{(f)}$	forecasted (aggregated) imports-exports, calendar information (weekday, hour of the day)
Static data $x_n^{(s)}$	node location (feeder to which the node is connected, and distance to the origin node)

end-users is leading to a high-dimensional output space, which may lead to high training complexity.

Here, we tackle this issue by using cross-series learning [30]. In this setting, we learn a single forecasting model f_θ (with parameters θ) which is shared by all end-users. Each user (using only its own private data along with publicly available information) predicts the voltage level corresponding to its node. In this way, clients with limited history will simply have fewer samples for training the joint model.

Overall, by learning from correlated voltage patterns from all individuals, the model f_θ acquires improved generalization abilities and is thus more robust to input noise (thus reducing overfitting risks), while accommodating clients with different histories (thus enabling cold-start forecasts for new clients joining the collaboration scheme). This strategy bypasses the need to retrain the model from scratch [31].

The model f_θ is used by each individual end-user $n \in \mathcal{N}$ for predicting, at the forecast creation time t_0, the conditional distribution of voltage levels $\mathbf{y}_{t_1:T,n} = (y_{t_1,n}, ..., y_{t_T,n})$ at node n over the horizon $[t_1, t_T]$ (using exclusively by information $\mathbf{x}_{t_0,n}^{\text{all}}$ available at node n before t_0):

$$f_\theta = \Pr\Big(\mathbf{y}_{t_1:T,n} \big| \underbrace{\mathbf{y}_{:t_0,n}, \mathbf{x}_{:t_0,n}^{(p)}, \mathbf{x}_{t_1:T,n}^{(f)}, x_n^{(s)}}_{\mathbf{x}_{t_0,n}^{\text{all}}}\Big) \ \forall n \in \mathcal{N} \quad (1)$$

where $\mathbf{y}_{:t_0,n}$ are the past nodal voltages (measured before t_0), $\mathbf{x}_{t_0:T,n}^{(p)}$ are the past time-varying covariates (before t_0), $\mathbf{x}_{t_1:T,n}^{(f)}$ are the known future covariates (over the horizon $[t_1, t_T]$), and $x_n^{(s)}$ are the time-invariant features. Note that the bold notation is used for vectors spanning over multiple time periods.

These covariates are summarized in Table I. It should be noted that the past net imports-exports of the community are publicly available, and provided by the central authority (e.g., distribution system operator). The imports predictions are obtained with an autoregressive model (using only past values). Moreover, we encode the position of each node in the grid, which enables to capture space discrepancies (from grid technical constraints) during the training phase. To that end, we use a generic approach, wherein we specify the feeder of the network to which the end-user is connected, along with the distance between the end-user and the feeder's root node. Overall, by feeding the model with both global (e.g., estimated imports-exports of the community and global PV conditions) and local (e.g., past consumption levels, position in the grid) features, the model is able to generalize to all end-users, while leveraging local information to adapt the predictions in regards to node-specific conditions (such as solar shading by trees).

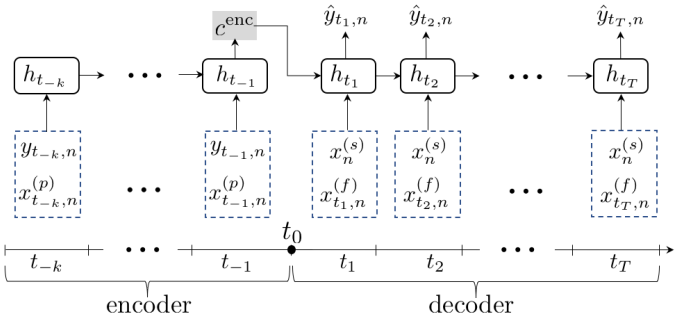

Fig. 1. Encoder-decoder model for solving the voltage forecasting problem.

B. Deep Learning Model Structure and Training

The multi-horizon time-series forecasting problem can be naturally treated as a sequence-to-sequence task, wherein the goal is to convert a sequence of k past observations into a sequence containing the T predictions of interest. In this work, this task is solved using an encoder-decoder model, i.e., an advanced deep learning model, which has shown high performance in different studies [32].

As shown in Fig. 1, the encoder processes past data over a look-back window of k time steps $[t_{-k}, t_{-1}]$, with the goal of extracting the relevant dynamics into a context vector c^{enc}. Then, at time t_0, the decoder leverages this vector c^{enc}, along with the known future data $\mathbf{x}_{t_1:T,n}^{(f)}$ and static features $x_n^{(s)}$, to generate the multi-horizon predictions $\hat{\mathbf{y}}_{t_1:T,n}$. Both encoder and decoder blocks are modeled by recurrent neural networks due to their ability to represent inter-temporal dependencies. Indeed, these models have an internal memory h_t that provides an internal representation of past events, which is used to propagate relevant information through time.

In this work, we are interested in probabilistic forecasts that return the full conditional distribution $p(\mathbf{y}_{t_1:T,n}|\mathbf{x}_{t_0,n}^{\text{all}})$. This target distribution is here approximated with Quantile Regression (QR) [33], which provides the conditional quantiles $\hat{y}_{t,n}^{(q)}\ \forall t \in [t_1, t_T]$, i.e., $p(y_{t,n} \leq \hat{y}_{t,n}^{(q)}|\mathbf{x}_{t_0,n}^{\text{all}}) = q$, for different relevant probability levels $q \in \mathcal{Q}$ [34]. For each node n, the QR model f_θ thus yields:

$$\hat{\mathbf{y}}_{t_1:T,n} = \left\{\hat{\mathbf{y}}_{t_1:T,n}^{(q)}\right\}_{q\in\mathcal{Q}} = f_\theta\left(\mathbf{x}_{t_0,n}^{\text{all}}\right) \ \ \forall n \in \mathcal{N} \quad (2)$$

where the output corresponding to each time step $t \in [t_1, t_T]$ is thus $|\mathcal{Q}|$-dimensional.

Practically, the model f_θ is trained on historical observations to learn the (unknown) relationship between inputs $\mathbf{x}_{t_0,n}^{\text{all}}$ and the outputs of interest $\hat{y}_{t,n}^{(q)}\ \forall t \in [t_1, t_T], n \in \mathcal{N}, q \in \mathcal{Q}$. This is achieved by adapting the model parameters θ so as to minimize a user-defined loss function $\mathcal{L}(\theta)$, which penalizes discrepancies between predictions and actual observations (in the training data). Here, we use the pinball loss since minimizing this metric yields the optimal quantiles of the forecast uncertainty [35]. This learning phase is carried out with a stochastic gradient descent (SGD) algorithm. In this iterative optimization process, one forms (at each round), a batch $b \in \mathcal{B}$ of $z_i = (\mathbf{x}_{t_0,n}^{\text{all}}, \mathbf{y}_{t_1:T,n})_i$ samples from the historical database,

and estimates (over these z_i samples) the gradient of the loss function $\mathcal{L}$ with respect to the θ-parameters as follows:

$$\nabla_\theta \mathcal{L}(\theta, b) = \frac{1}{|b|} \sum_{i=1}^{|b|} \nabla_\theta \mathcal{L}(\theta, z_i) \tag{3}$$

where $\nabla_\theta \mathcal{L}(\theta, z_i) = \left[\frac{\partial \mathcal{L}(\theta, z_i)}{\partial \theta_1}, ..., \frac{\partial \mathcal{L}(\theta, z_i)}{\partial \theta_{|\theta|}}\right]$. Then, the weight vector θ is updated following the direction of the batch-averaged gradient $-\nabla_\theta \mathcal{L}(\theta, b)$ towards a local minimum, i.e.,

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta, b) \tag{4}$$

where η is the learning rate. It should be noted that SGD variants have been developed to reach better performances. One popular example is the Adam algorithm [36] that relies on adaptive learning rates to improve convergence properties.

C. Federated Learning

Federated learning (FL) is a distributed approach where a federation of clients $c \in \mathcal{C}$ is coordinated by a central server to learn a global model f_θ, without sharing any raw client data.

In [21], an innovative algorithm, called Federated Averaging, is developed. In this setting, the server initializes the parameters θ_0. At each round $r \in \mathcal{R}$ of the federated training, a sample of clients $\mathcal{C}_r \subseteq \mathcal{C}$ is selected, to whom the server broadcasts the global model θ_{r-1}. Then, each selected client $c \in \mathcal{C}_r$ performs local computations (e.g., SGD or Adam) on its private dataset, and computes the difference $\Delta_{c,r}$ between the new (locally) optimized model $\theta_{c,r}$ and the global model θ_{r-1}, i.e., $\Delta_{c,r} = \theta_{c,r} - \theta_{r-1}$. The local updates $\Delta_{c,r}\ \forall c \in \mathcal{C}_r$ are then uploaded to the server, which calculates the global average (5), i.e.,

$$\Delta_r = \frac{1}{|\mathcal{C}_r|} \sum_{c \in \mathcal{C}_r} \Delta_{c,r} \tag{5}$$

where $|\mathcal{C}_r|$ is the number of clients in training round r. The new global model is then computed as

$$\theta_r \leftarrow \theta_{r-1} + \eta_s \Delta_r \tag{6}$$

with η_s the learning rate at the server.

It should be noted that the Federated Averaging algorithm enables to perform multiple local updates, e.g., multiple steps of the SGD (3)-(4), before the averaging step (5). This reduces the number of communication iterations between the server and the clients. As further discussed in subsection III-C, by limiting these interactions (during which an adversary can potentially access the parameters θ_r), stronger privacy guarantees can be achieved by the final model.

D. Differential Privacy

During the federated learning, the model can be accessed by adversaries to infer raw local information. Encrypted computations can be used to protect the training procedure, but there are still open questions about the possibility to break current cryptographic functions. Moreover, encryption schemes are computationally expensive, and consume large amounts of energy, which conflicts with the goal of energy efficiency.

An emerging alternative is provided by differential privacy (DP), which enables to bound and quantify the privacy leakage of sensitive information when performing learning tasks [29]. DP offers provable guarantees of protection against adversaries that have full knowledge of the training procedure along with an access to the model's parameters. DP is based on the notion of adjacent databases D and D', which differ by the addition or removal of a single element. In this work, we consider user-level DP, which focuses on the largest possible difference that one client can have on the trained model [37].

In the context of deep learning, $\mathcal{M}$ refers to the learning algorithm (e.g., gradient descent optimization), and the application of $\mathcal{M}$ on a database D yields the model weights $\theta = \mathcal{M}(D)$. Since $\mathcal{M}$ is inherently stochastic (e.g., neural networks are initialized with random weights, and rely on a random selection of samples at each weights update), there is uncertainty on the final weights θ of the deep learning model. Formally, a randomized learning algorithm $\mathcal{M}$ is said to be (ϵ, δ)-differentially private if, for any adjacent datasets D and D', and any subset S in the weights distribution, we have:

$$\Pr(\underbrace{\mathcal{M}(D)}_{=\theta_1} \in S) \leq e^\epsilon \Pr(\underbrace{\mathcal{M}(D')}_{=\theta_2} \in S) + \delta \tag{7}$$

where $\epsilon \geq 0$ is the privacy loss yielding an upper bound of how much the probability of converging to a particular set of weights θ is affected by including (or removing) a single client during training $\mathcal{M}$. A low ϵ-value is preferable since it means that removing any end-user from the training database does not significantly modify the final model. Such a property makes it very difficult to infer the raw data of clients. Then, we see that (ϵ, δ)-DP allows for potentially large privacy losses (no bound on ϵ) with probability δ. In this way, $\delta \in [0, 1]$ is the failure probability which caps any long tail of the $\mathcal{M}(\cdot)$-distribution where pure ϵ-DP guarantees do not hold. In the worst-case scenario wherein this δ-fraction exclusively relates to a single client, this may prove detrimental. Hence, to ensure privacy for each end-user $n \in \mathcal{N}$, a solution is to have $\delta < 1/|\mathcal{N}|$.

Overall, a training process $\mathcal{M}$ is differentially private if the probability of $\theta_1 = \mathcal{M}(D)$ and $\theta_2 = \mathcal{M}(D')$ are close for every choice of D and D', i.e., the data of any client do not significantly affect the weights distribution of the algorithm. In this work, the goal is to convert the federated learning of neural networks into a differentially private distributed training $\mathcal{M}$ that is associated with formal (i.e., provable) privacy guarantees (by bounding ϵ and δ values).

III. Differentially Private Federated Forecaster

In this section, we present how to train deep learning models with user-level DP. The methodology is summarized in Fig 2. First, the forecasting model is initialized (i.e., training round $r = 1$) at the server side, and sent to a random subset $\mathcal{C}_{r=1}$ of end-users. Each client $c \in \mathcal{C}_{r=1}$ locally trains the model (using its own private data and public information). Then, local updates $\Delta_{c,r}$ are sent back to the server, and are aggregated with the addition of noise that is calibrated using DP in such a way that the privacy leakages can be bounded and quantified. This procedure is iterated (over training rounds $r \in \mathcal{R}$) until

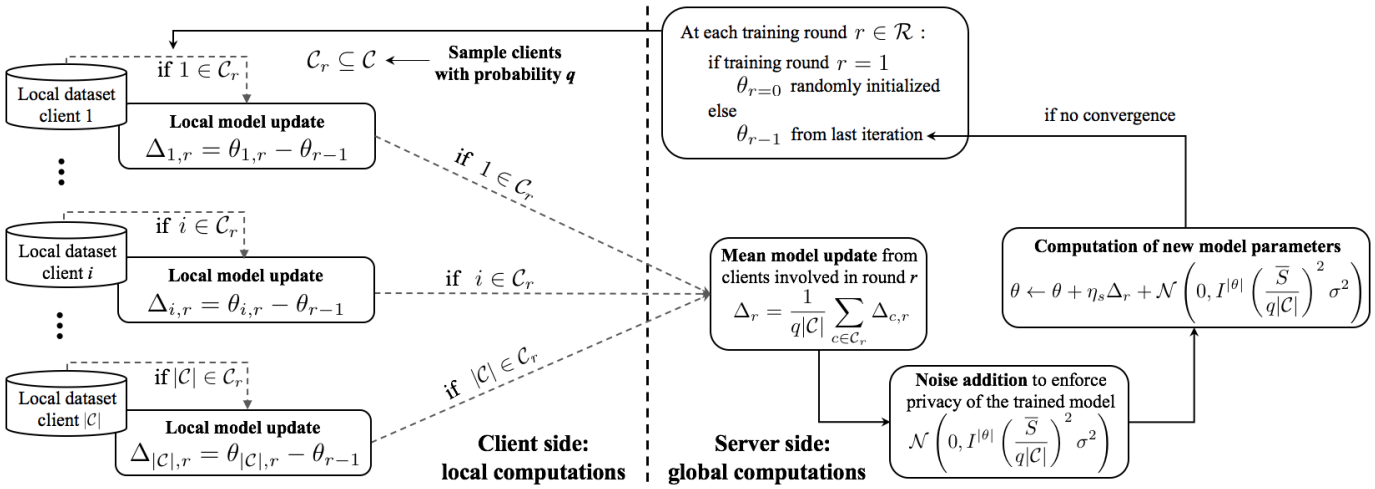

Fig. 2. Overview of the proposed privacy-preserving probabilistic forecasting tool.

convergence, i.e., when the model accuracy cannot be further improved or when the privacy budget is consumed.

The core of the methodology is explained in subsection III-A. It is then enriched in subsection III-B with the introduction of a new privacy-compliant procedure that internalizes the data normalization (of individual end-users) into the prediction model, which improves consistency among the different local updates $\Delta_{c,r}$. Finally, we compute in subsection III-C the total privacy cost of the trained model.

A. Differentially Private Deep Federated Learning

Different approaches have recently been proposed for the differentially private training of learning models [38], [39]. As discussed in [29], an effective solution to ensure that a learning function f achieves (ϵ, δ)-DP is to add noise proportional to the sensitivity S_f of that function:

$$\mathcal{M}(D) = f(D) + Noise(S_f) \tag{8}$$

where S_f is defined as the maximum of the l_2-distance $\|f(D) - f(D')\|_2$ for any adjacent input datasets D and D'. In particular, it is shown in [40] that a function f with sensitivity S_f can achieve (ϵ, δ)-DP by adding Gaussian noise $\mathcal{N}(0, S_f^2\sigma^2)$, with $\epsilon \leq 1$ and $\delta \geq 0.8 \cdot \exp\left(-(\sigma\epsilon)^2/2\right)$. The resulting learning process is usually referred to as Gaussian mechanism, i.e., $\mathcal{M}(D) \triangleq f(D) + \mathcal{N}(0, S_f^2\sigma^2)$, where σ is the noise multiplier controling the trade-off between privacy and model performance.

In the context of (centrally trained) neural networks, DP can thus be achieved by applying the Gaussian mechanism to the weight update function (4) of the stochastic gradient descent (SGD) algorithm [41], i.e.,

$$\theta \leftarrow \theta - \eta \frac{1}{|b|} \left(\sum_{i=1}^{|b|} \nabla_\theta \mathcal{L}(\theta, z_i) + \mathcal{N}\left(0, I^{|\theta|} S_f^2 \sigma^2\right) \right) \tag{9}$$

where $|b|$ is the number of samples z_i in the SGD mini-batch $b \in \mathcal{B}$, and $I^{|\theta|}$ is an identity matrix of size $|\theta|$ such that $\mathcal{N}(0, I^{|\theta|} S_f^2 \sigma^2)$ is a $|\theta|$-dimensional Gaussian noise.

This principle can be further extended to FL [37], by using a DP version of the global average update (5), i.e., where per-client updates $\Delta_{c,r}$ are aggregated (on the server) at each round r of the training:

$$\theta \leftarrow \theta + \eta_s \frac{1}{|\mathcal{C}_r|} \left(\sum_{c \in \mathcal{C}_r} \Delta_{c,r} + \mathcal{N}\left(0, I^{|\theta|} S_f^2 \sigma^2\right) \right) \tag{10}$$

However, in traditional SGD algorithms, the sensitivity S_f of the update function f is a priori unknown. A solution is proposed in [41], wherein the gradient of each sample is clipped at a threshold $\overline{S}$ before the update step, such that the maximum influence (i.e., sensitivity) of a sample on the final average is bounded, i.e., $S_f \leq \overline{S}$.

However, this technique only offers privacy for single samples. In order to extend sample-level DP to user-level DP, we need to bound the sensitivity of the training function (10) with respect to the addition or removal of any client from the dataset. This is accomplished by clipping the local weights updates $\Delta_{c,r}$ (at the end of the local training round r of each end-user c). In particular, for each user, we bound the (l_2-norm of) local updates $\Delta_{c,r}$ by $\overline{S}$, i.e., $\|\Delta_{c,r}\|_2 \leq \overline{S}$, as follows:

$$\Delta_{c,r} \leftarrow \Delta_{c,r} \cdot \min\left(1, \frac{\overline{S}}{\|\Delta_{c,r}\|_2}\right) \tag{11}$$

such that if $\|\Delta_{c,r}\|_2 > \overline{S}$, per-user updates $\Delta_{c,r}$ are reduced to $\overline{S}$, while $\Delta_{c,r}$ are preserved otherwise. It should be noted that bounding the influence of any user is also beneficial for training stability since it prevents the model to overfit to a particular subset of data.

When considering privacy in the federated learning of deep networks, it is important to notice that privacy breaches may occur at each round r between clients and the server, through the information contained in the weight updates. To mitigate this issue, we exploit the randomness associated with subsampling. Indeed, if the training $\mathcal{M}$ is (ϵ, δ)-DP, then drawing a random subset of end-users (from all $|\mathcal{C}|$ clients) before applying $\mathcal{M}$ follows $(\mathcal{O}(\gamma\epsilon), \gamma\delta)$-DP, with $\gamma < 1$ (which depends on the sampling strategy) [42].

Here, end-users are randomly and independently sampled with probability $q \in (0, 1]$ at each round $r \in \mathcal{R}$ of the training mechanism $\mathcal{M}$. Hence, the number $|\mathcal{C}_r|$ of end-users at each round r is variable and unknown. It is thus replaced by its expected value $\mathbb{E}[|\mathcal{C}_r|] = q|\mathcal{C}|$, such that the global model update (on the server) is:

$$\theta \leftarrow \theta + \eta_s \frac{1}{q|\mathcal{C}|} \sum_{c \in \mathcal{C}_r} \Delta_{c,r} + \mathcal{N}\left(0, I^{|\theta|}\left(\frac{\overline{S}}{q|\mathcal{C}|}\right)^2 \sigma^2\right) \quad (12)$$

where the sensitivity S_f of the update function (12) is bounded by $S_f \leq \frac{\overline{S}}{q|\mathcal{C}|}$, and σ is the noise multiplier for training the neural network.

The resulting differentially private federated learning for deep networks (DP-FDL) is given in Algorithm 1. By combining sampling and additive Gaussian noise to the update function, the training procedure $\mathcal{M}$ follows a Sampled Gaussian mechanism (SGM), which is sufficient for achieving a central differential privacy (section III-C).

Algorithm 1 Differentially private federated learning for deep networks (DP-FDL)

function SERVERTRAIN($q, \sigma, \eta_c, \eta_s, \overline{S}$)
  Initialize model θ_0
  for each round $r \in \mathcal{R}$ **do**
    $\mathcal{C}_r \leftarrow$ (users sampled with probability q)
    for each client $c \in \mathcal{C}_r$ **do**
      $(\Delta_{c,r}) \leftarrow$ LOCALTRAIN(c, θ_{r-1}, η_c, $\overline{S}$)
    end for
    $\tilde{\Delta}_t = \frac{1}{q|\mathcal{C}|} \sum_{c \in \mathcal{C}_r} \Delta_{c,r} + \mathcal{N}\left(0,\, I^{|\theta|}\left(\frac{\overline{S}}{q|\mathcal{C}|}\right)^2 \sigma^2\right)$
    $\bar{\Delta}_t = \beta \bar{\Delta}_{t-1} + \tilde{\Delta}_t$
    $\theta_{r+1} \leftarrow \theta_r + \eta_s \bar{\Delta}_t$
  end for
end function

function LOCALTRAIN(c, θ_{r-1}, η_c, $\overline{S}$)
  $\theta \leftarrow \theta_{r-1}$
  split local data into batches $b \in \mathcal{B}$
  for each batch $b \in \mathcal{B}$ **do**
    $\theta \leftarrow \theta - \eta_c \nabla_\theta \mathcal{L}(\theta, b)$
  end for
  $\Delta \leftarrow \theta - \theta_{r-1}$
  return $\left(\Delta \cdot \min\left(1, \frac{\overline{S}}{\|\Delta\|_2}\right)\right)$
end function

As shown in Algorithm 1, federated averaging is augmented with server momentum β, due to its ability to dampen oscillations in the learning, and thus to mitigate that local models diverge from the globally optimal solution.

B. Features Normalization

In general, end-users have different distributions of data. In a centralized learning, this poses no problem as a common normalization of features is applied over the entire database. However, when local data are private, such a shared computation cannot be performed. This prevents the use, e.g., of traditional batch normalization [43] wherein local statistics (i.e., mean and variance values) are aggregated over the whole training data (including all clients) to generate the predictions.

To address this issue, we use layer normalization (LN) [44]. In contrast to traditional approaches, LN does not rely on shared statistics among end-users (that would break data privacy), and rather internalizes the normalization into the first layer of the model. This generic procedure can thus be applied to any neural network architecture. For the inputs $\mathbf{x}_i$ in a neural layer, the LN inputs $\mathbf{x}_i^{\text{nrm}}$ are given by:

$$\mathbf{x}_i^{\text{nrm}} = f\left(\frac{\mathbf{g}}{\sigma_i} \odot (\mathbf{x}_i - \mu_i) + \mathbf{b}\right) \quad (13)$$

where $\odot$ is the element-wise multiplication of vectors. All units in the layer thus share the same normalization terms μ_i and σ_i, i.e., the mean and standard deviation of the elements in $\mathbf{x}_i$, which are sample-dependent and computed locally. The bias $\mathbf{b}$ and gain $\mathbf{g}$ vectors are the internal parameters of the model that need to be learned during training. These vectors $\mathbf{b}$ and $\mathbf{g}$ are thus common to all clients, such that the normalization procedure (involving all clients) is directly internalized into the privacy-preserving learning.

C. Privacy Budget

The DP-FDL algorithm consists of $|\mathcal{R}|$ successive queries of the learning function $\mathcal{M}$ in Eq. (12). An adversary with access to intermediate models θ_r may leverage this additional information to infer raw (private) data, and it is thus necessary to keep track of the total privacy loss, which is accumulating during the training $(\mathcal{M}_1, \ldots, \mathcal{M}_{|\mathcal{R}|})$.

A solution is offered by advanced composition theorems [45], which give an upper bound of the accumulated privacy loss (by assuming that the worst-case scenario wherein the same amount of leakage occurs at each query to the data). For instance, applying $|\mathcal{R}|$ times consecutively the same (ϵ, δ)-DP algorithm gives an $\left(\mathcal{O}\left(\epsilon\sqrt{|\mathcal{R}|\log(1/\delta)}\right), |\mathcal{R}|\delta\right)$ guarantee [46]. However, these compositions are generic (they do not account for the specific noise distribution used in the learning), such that they tend to strongly exaggerate privacy losses.

This has motivated the development of another approach, called the moments accountant [41], which, instead of directly dealing with (ϵ, δ)-DP, relies on the notion of Rényi-DP (RDP) [47]. It is a natural relaxation of DP based on the Rényi divergence (14) of order $\alpha > 1$ between distributions P and Q (defined over the same probability space):

$$\mathrm{D}_\alpha(P \parallel Q) = \frac{1}{\alpha - 1} \log \mathbb{E}_{z \sim Q}\left(\frac{P(z)}{Q(z)}\right)^\alpha \quad (14)$$

where $\log$ is the natural logarithm, and $P(z)$ is the density of P at z. By definition, $\mathcal{M}$ satisfies (α, γ)- RDP if for any adjacent inputs D, D', it holds that $\mathrm{D}_\alpha(\mathcal{M}(D) \parallel \mathcal{M}(D')) \leq \gamma$.

The main advantage of RDP is its simple linear composition form, i.e., if $\mathcal{M}$ obeys (α, γ)-RDP, then the composition $\mathcal{M}_{|\mathcal{R}|}$ is $(\alpha, \gamma|\mathcal{R}|)$-RDP [47]. In particular, a numerically stable computational procedure for estimating the Rényi parameters for a sampled Gaussian mechanism is presented in [48], and has shown to provide strengthened privacy bounds.

However, in contrast with (ϵ, δ)-DP, Rényi parameters are more difficult to interpret, and we are thus interested in

converting the privacy budget expressed in terms of (α, γ) in the more interpretable notion of (ϵ, δ)-DP. In Proposition 3 in [47], it is shown that an (α, γ)-RDP algorithm provides $\left(\gamma + \frac{\log(1/\delta)}{\alpha-1}, \delta\right)$-DP guarantees. However, follow-up works have improved this conversion, and bounds provided by [49] are therefore used in our experiments (in section V-B) to tighten the privacy guarantees of the learning procedure.

It should be noted that few studies focus on how selecting an adequate privacy budget ϵ, although it is well-established that higher privacy budgets can be allocated to more complex learning tasks [39]. In that regard, it has been shown that DP-enhanced neural networks are resistant to inference attacks (with state-of-the-art attack frameworks) for ϵ up to 100 [50]. In complement, studies on memorization attacks (which exploit the fact that highly parametrized models may memorize some patterns in training data) demonstrated that attacks have been blocked by DP mechanisms with poor worst-case privacy guarantees (high ϵ values up to 10^9) [51]. In section V-B, the trade-off between data privacy and model performance is fully discussed.

IV. Baselines

In section IV-A, we introduce baseline models (not trained in a privacy-aware setting) to evaluate the forecasting accuracy of state-of-the-art techniques in traditional (non-private) conditions. Then, in section IV-B, we present the deep learning models that are trained in both (non-private) conditions and with the proposed DP-enhanced federated learning. Finally, we present the performance metric used to compute the prediction performance of the different models in section IV-C.

A. Baseline Models

We implement naive and state-of-the-art techniques for the multi-horizon forecasting task.

- a naive probabilistic averaging model (Prob-Avg), where the voltage distribution at each time step is computed based on the aggregation of all past observations corresponding to this specific period.
- a naive probabilistic persistence (Prob-Persistence), where the last nodal voltage value $y_{t_{-1},n}$ is propagated over the prediction horizon $[t_1, t_T]$ as the mean value, while the variance is computed on the look-back window.
- a quantile regression forest (QRF), i.e., a tree-based ensemble model, in which the outputs of independent regression trees are merged for estimating the conditional distribution. The forest is set to 500 trees.
- a gradient boosting regression tree (QGBRT) trained with the quantile loss. In this ensemble model, new regression trees are sequentially generated to forecast the residuals of the previous models. The number of boosting stages is set to 100 with an early stopping criterion.

B. Deep Learning Models

The sequence-to-sequence model (of section II-B) is tested with Long Short Term Memory (LSTM) recurrent networks for both encoder and decoder blocks. A second variant (BLSTM) relies on bidirectional LSTM networks to improve the representation of time dependencies [34].

We also use a traditional deep feedforward neural network (DFFNN), wherein hidden layers are composed of neurons using rectifier linear units (ReLUs) as activation function.

In complement, we build a BLSTM model wherein the local (private) features of all end-users are aggregated into a single input vector to jointly predict the voltage levels at all nodes of the grid. By constructing a shared input space, the total number of training samples is limited by the client with the smallest database. This model (A-BLSTM) serves thus as a benchmark to quantify the value of the cross-learning strategy (wherein all historical data are considered by treating each client as a different sample).

C. Hyper-parameters Optimization and Performance Metrics

The input features of all machine learning (tree-based and deep learning) models are the same (as given in Table I). This strategy is used to ensure that differences in prediction performance only arise from privacy constraints.

For each forecasting model (excepting the parameter-free naive methods), an hyperparameter optimization is carried out through an extensive random search to identify the optimal model complexity. The same number of iterations is used across all benchmarks.

When assessing the performance of a probabilistic forecast, two complementary aspects need to be jointly analyzed, i.e., reliability and sharpness.

Reliability measures how closely the predicted intervals correspond to the actual data frequencies, while sharpness measures the width of prediction intervals. To evaluate the trade-off between both concepts, we use the quantile loss. It has indeed been shown the quantile loss yields consistent outcomes with other metrics such as the Winkler score, and the continuous ranked probability score (CRPS) [52].

The quantile loss $QL_{\tau,n}$ for node n at time step τ of the test set is given by:

$$QL_{\tau,n} = \sum_{q \in \mathcal{Q}} q \max\left(y_{\tau,n} - \hat{y}^{(q)}_{\tau,n}, 0\right) + (1-q)\max\left(\hat{y}^{(q)}_{\tau,n} - y_{\tau,n}, 0\right) \tag{15}$$

where $\hat{y}^{(q)}_{\tau,n}$ are the quantiles predicted by the forecaster, while $y_{\tau,n}$ are the actual voltage observations. In this paper, we compute the quantiles for q = 1, 10, 25, 50, 75, 90 and 99 %.

Practically, we compute the total pinball loss QL^{tot}, i.e., the average value of all pinball losses (15) over all points (t, n) of the space-time domain of the test set. Smaller values of QL^{tot} correspond to better forecasting outcomes.

V. Case Study

The proposed privacy-preserving voltage forecasting strategy is tested on the IEEE European Low Voltage Test Feeder, shown in Fig. 3, which has a radial structure with 6 line ramifications (used to encode the spacial information of each node). The nodal voltages are predicted over a multi-horizon of T = 8 intervals of 30 minutes (i.e., 4-hour ahead) for the

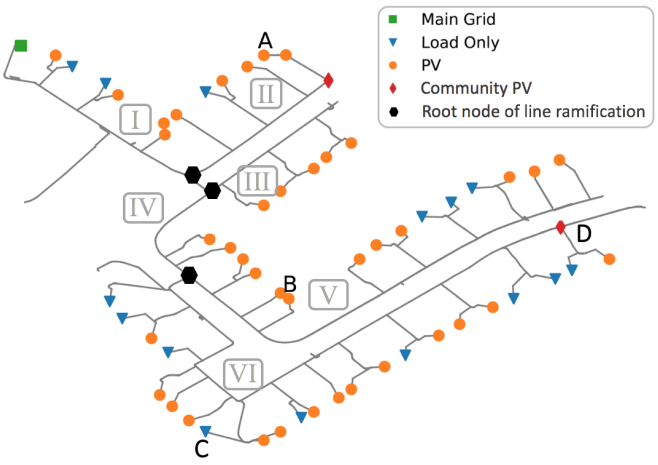

Fig. 3. The studied energy community, composed of 6 line ramifications (I to VI) feeding 57 clients, i.e. 15 single-phase inflexible loads, 40 single-phase prosumers (with inflexible load, PV source and battery system) and 2 three-phase community-scale PV plants.

N = 57 nodes. A look-back window of k = 12 intervals (i.e., 6 hours) is selected to capture past dynamics. Each client has a different history of recordings, ranging from 120 to 478 days. To have comparable performance indices among end-users, the test set is composed of the same days for all clients (corresponding to the last 96 days of the horizon). This test set is used to compute performance metrics that are independent from the data used during the model training. From these remaining data (not used in the test set), 85% are employed for training the model, while the remaining 15% are used for tuning the hyper-parameters. Overall, the aggregated test set is thus composed of 96×N = 96×57 = 5,472 daily voltage profiles, while the training and validation sets respectively include 11,400 and 1,995 daily patterns.

In this case study, a significant dependency between voltage levels and local energy exchanges is observed. In particular, using the Pearson coefficient [53], the nodal voltages have a correlation of -0.51 with the consumption profile, and of 0.64 with the PV generation. The voltage levels provide thus an indirect representation of local consumption and generation data (that reveal critical and private information), and it is essential to protect the privacy of the voltage forecaster.

A. Comparison Between Centralized and Local Models

To quantify the practical interest for end-users to participate in collaborative learning strategies, we compare the forecasting performance of centralized and fully private models.

First, we study the prediction performance of (non-private) centralized models relying on complete information, i.e., an ideal (but unrealistic) case where each end-user shares its private data to form a single database, from which the community manager trains a single model.

Second, we implement the fully private counterpart, where a different model is trained by each end-user based only on its own private data.

In deep learning-based models, the weights are initialized using a Glorot uniform distribution, while the batch size is set to 96 (as a good trade-off between training time and final performance). The optimization is performed with the Adam algorithm with a learning rate of 0.001. Then, in all tree-based and deep learning models, early stopping is implemented for avoiding overfitting in the learning procedure.

Table II presents the performance and training time of the different probabilistic forecasters. For deep learning model, the number of epochs is added in parentheses. It should be noted that, in the private setting, the performance of local models is averaged over all end-users of the community.

TABLE II
PERFORMANCE AND TRAINING TIME OF DIFFERENT MODELS.

Model	Centralized		Local (private)	
	QL^{tot} [pu]	time [min] (epochs)	QL^{tot} [pu]	time [min] (epochs)
Prob-Persist	-	-	0.210	-
Prob-Avg	0.319	-	0.192	-
QRF	0.048	148	0.156	172
QGBDT	0.041	64	0.151	76
DFFNN	0.043	10 (33)	0.144	16 (27)
LSTM	0.034	52 (18)	0.186	63 (4)
BLSTM	0.032	88 (16)	0.172	105 (4)
A-BLSTM	0.192	54 (11)	-	-

The experiments show that machine learning models (deep learning and ensemble methods) significantly outperform naïve benchmarks (in both centralized and local settings), which stresses the interest of using advanced models to properly represent the dynamics of nodal voltage levels. In particular, centralized BLSTM-based networks (with 1 hidden layer composed of 50 neurons for both encoder and decoder blocks) achieve the best accuracy with QL^{tot} = 0.0032 pu, due to their ability to capture the non-linear inter-temporal dependencies within data. The convergence is obtained in 16 rounds, with a per-round training time of around 5.5 minutes.

Moreover, we observe high differences in performance between centralized and local models. Specifically, the local (private) probabilistic averaging (Prob-Avg) wherein the voltage distributions are differentiated between nodes, reduces the total quantile loss of the centralized version from 0.319 pu down to 0.192 pu. This difference reflects the importance of capturing nodal dependencies between voltage levels.

Surprisingly, when training fully private models (using only local data), sequence-to-sequence networks exhibit poor convergence, and are even surpassed by tree-based and feed-forward networks. This arises from the difficulty to train recurrent neural networks from scratch with a limited amount of data. In this way, the hyper-parameter optimization has led to very compact LSTM (and BLSTM) local models (with 1 hidden layer of 8 neurons). Also, the A-BLSTM (where all nodal voltage levels are jointly predicted) only yields QL^{tot} = 0.192 pu, which can be explained by the data paucity (i.e., the number of samples is limited by the client with only 120 days of measurements). In this way, by enlarging the dataset (with the aggregation of samples from all end-users $n \in \mathcal{N}$), the cross-learning approach reduces the uncertainty space of the predictions, leading to improved sharpness. This materializes by a total quantile loss (for the centralized BLSTM) of 0.0032 pu over the test set.

In contrast, the best local model (DFFNN) performs no better than QL^{tot} = 0.0144 pu. This performance gap between centralized and local models is a clear indication of the added value for end-users to collaboratively train a community-wide probabilistic forecaster. This need is exacerbated for end-users with data paucity, as they are particularly exposed to poor convergence and weak accuracy.

B. Utility – Privacy Trade-off

There is an inherent trade-off between the performance of the prediction model and the privacy, i.e., (ϵ, δ)-values, of the local data. This trade-off is affected by three main parameters. First, privacy breaches may occur at each round r of the training, i.e., stronger privacy guarantees are obtained when the training phase is quickly converging. Second, privacy of raw data can be improved by reducing the expected number of end-users $q|\mathcal{C}|$ at each round. However, in accordance with Eq. (12), reducing $q|\mathcal{C}|$ is increasing the sensitivity $\left(\frac{\overline{S}}{q|\mathcal{C}|}\right)$ of the update function. In turn, this augments the variance of the noise added during the server optimization, which is detrimental for prediction accuracy. Third, the noise multiplier σ is controling the variance of the noise injected during the weight update of the learning phase. Higher σ-values improve privacy guarantees, but at the expense of accuracy.

Here, we quantitatively analyze the utility-privacy trade-off. We study three different deep learning models, along with the BLSTM without layer normalization (called I-BLSTM) wherein the inputs of each end-user are normalized independently (based only on local statistics). The models are trained during $|\mathcal{R}|$ = 60 rounds. As explained in section II-D, to ensure $\delta < 1/|\mathcal{N}|$, the ϵ values are reported for $\delta = 10^{-2}$ in the following of the paper.

In Table III, we give the optimal number of rounds r^{opt}, i.e., the number of interactions between end-users and the server which leads to the optimal accuracy on the validation set. Then, we provide the corresponding privacy loss ϵ^{opt} (using Proposition 3 in [44], as explained in section III-C), along with the total quantile loss (over the test set) for two noise levels (σ = 0.25 and 0.75) for $q|\mathcal{C}|$ = 10 users per round. The Adam algorithm with η_c = 0.001 is used for training local models, while SGD with a learning rate η_s = 1 is used at the server side. The momentum β is set at 0.6 in Algorithm 1.

TABLE III
PERFORMANCE OF DIFFERENT PRIVACY-PRESERVING MODELS.

Model	$q\|\mathcal{C}\|$	σ	r^{opt}	ϵ^{opt}	QL^{tot} [pu]
DFFNN	10	0.25	53	173.6	0.096
LSTM	10	0.25	56	182.6	0.089
BLSTM	10	0.25	54	167.7	0.090
I-BLSTM	10	0.25	54	167.7	0.115
DFFNN	10	0.75	47	9.1	0.140
LSTM	10	0.75	56	10.0	0.117
BLSTM	10	0.75	32	7.3	0.112
I-BLSTM	10	0.75	48	9.2	0.135

From Table III, we see that (LSTM and BLSTM) sequence-to-sequence models are more robust to the noise added, while the DFFNN struggles in achieving decent outcomes in the DP framework. In this way, while the BLSTM is still able to reach QL^{tot} = 0.112 pu for $\sigma = 0.75$, the DFFNN only performs 0.140 pu (which is roughly equivalent to the private models). Also, outcomes stress the added value of using layer normalization in the federated learning since the I-BLSTM model is consistently outperformed by the reference BLSTM.

Overall, raising σ from 0.25 to 0.75 significantly reduces the worst-case privacy loss, but this comes at the expense of forecasting performance (for all models). It should be noted that using a noise multiplier higher than 0.8 even leads to strong model divergence. To better understand the effect of the noise level σ and the expected number of clients per round $q|\mathcal{C}|$ on the utility-privacy trade-off, Table IV shows outcomes from a sensitivity analysis on the performance of the BLSTM model in different privacy settings.

TABLE IV
PERFORMANCE OF THE BLSTM FOR DIFFERENT PRIVACY SETTINGS.

Model	$q\|\mathcal{C}\|$	σ	r^{opt}	ϵ^{opt}	QL^{tot} [pu]
BLSTM	5	0	59	-	0.060
BLSTM	5	0.25	60	105.1	0.104
BLSTM	5	0.75	38	4.0	0.132
BLSTM	10	0	57	-	0.054
BLSTM	10	0.25	54	167.7	0.090
BLSTM	10	0.75	32	7.3	0.112

In Table IV, we see that models trained with traditional federated learning (which is not augmented with differential privacy, i.e., σ = 0) cannot reach the performance of centralized models, but they are still significantly better than their private counterpart. In particular, the federated BLSTM with $q|\mathcal{C}|$ = 10 end-users per round converges towards QL^{tot} = 0.054 pu, while the centralized and private equivalents respectively achieve 0.032 pu and 0.172 pu.

Interestingly, the subsampling strategy allows to achieve more stringent privacy guarantees, i.e., lower ϵ-values are obtained for $q|\mathcal{C}|$ = 5 than for $q|\mathcal{C}|$ = 10 for the same number of training rounds. However, in accordance with Eq. (12), decreasing the expected number of clients per round $q|\mathcal{C}|$ is increasing the sensitivity $\left(\frac{\overline{S}}{q|\mathcal{C}|}\right)$ of the update function. In turn, this augments the variance of the noise added during the server optimization, which is detrimental for prediction accuracy. This effect is clearly observed with the performance gaps between $q|\mathcal{C}|$ = 5 and 10 (e.g., QL^{tot} increases from 0.112 pu up to 0.132 pu for σ = 0.75). In our case, further increasing $q|\mathcal{C}|$ to 12 or 15 end-users does not improve the results, and we conclude that the best solution is to use $q|\mathcal{C}|$ equal to 10. For illustrating the quality of results obtained using the BLSTM network with $q|\mathcal{C}|$ = 10 and σ = 0.25, the probabilistic voltage forecasts of 4 nodes (A, B, C and D in Fig. 3) during a summer day are shown in Fig. 4. The gray areas represent the forecasted quantiles while the red line stands for the actual voltage time series.

Fig. 4 shows that the predicted intervals properly encapsulate the actual voltage realizations, i.e., the volatility of nodal voltages is well captured in tight intervals. To have better insights on the convergence of the DFFNN, BLSTM and I-BLSTM models, their training performances are illustrated in Fig. 5. We depict (on a logarithmic scale) the evolution of the total quantile loss QL^{tot} over the course of training for three

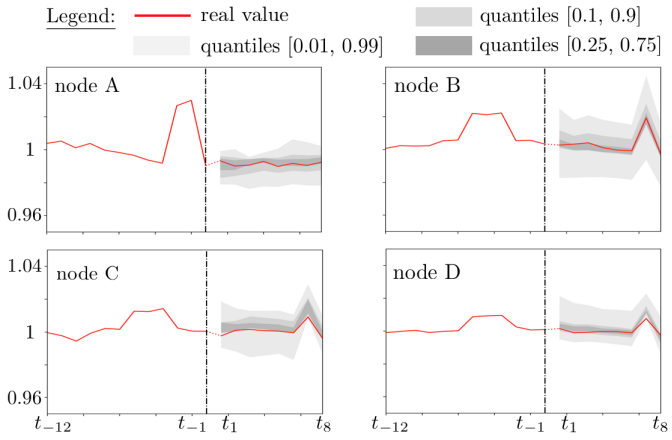

Fig. 4. Probabilistic voltage forecasts for four different nodes of the low-voltage community.

levels of the noise multiplier (σ = 0, 0.25, and 0.75) for $q|\mathcal{C}|$ = 10. We also report the corresponding evolution (at each round $r \in \mathcal{R}$) of the privacy loss ϵ (with $\delta = 10^{-2}$).

In absence of noise, the DFFNN converges faster than the sequence-to-sequence BLSTM model, which arises from its simpler architecture that is easier to optimize. In contrast, the BLSTM model exhibits a more gradual training but is ultimately able to converge towards better solutions, i.e., the BLSTM outperforms the DFFNN for all noise levels σ. Also, we see that the DFFNN is more sensitive to noise than the BLSTM model, i.e., even a small noise value $\sigma = 0.25$ has a substantial effect on the convergence abilities of the DFFNN. The BLSTM is thus more suited to be used in combination with DP. Finally, the value of the layer normalization occurs at the end of the training (when the search towards the optimal solution becomes finer), by allowing a more efficient transfer of learning between end-users.

C. Sensitivity Analysis on Hyper-parameters

In addition to the parameters specific to privacy, the prediction accuracy of the forecasters is also affected by other hyper-parameters, which need to be carefully tuned. In Table V, we evaluate the influence of the number of hidden units (# neurons), batch size ($|b|$) and clipping threshold ($\overline{S}$) on the BLSTM test performance and per-round training time for σ = 0.25 and $q|\mathcal{C}|$ = 10, thus deriving valuable insights for applying differentially private federated learning in smart grid forecasting applications.

TABLE V
SENSITIVITY ANALYSIS ON THE BLSTM PERFORMANCE.

# neurons	$\|b\|$	$\overline{S}$	QL^{tot} [pu]	time [sec]
10	96	0.175	0.090	41
5	96	0.175	0.095	38
50	96	0.175	0.150	52
10	10	0.175	0.091	123
10	200	0.175	0.099	32
10	96	0.1	0.111	40
10	96	0.3	0.163	42

In contrast with centralized training wherein more complex models have a higher modeling power, using more hidden units in the federated DP framework strongly decreases the model quality. In practice, it is thus preferable to use more compact models, which are less sensitive to the noise added in the learning procedure. This has also a positive effect on the communication burden since it requires transferring a smaller weight vector (and thus less bandwidth) between the server and end-users.

Then, the batch size has a very small effect on the final performance but it significantly influences the computation time. In this way, by decreasing the batch size from 96 to 10, the per-round training time increases from 41 to 123 seconds.

Finally, identifying the optimal value of the clipping threshold $\overline{S}$ is not an easy task. Indeed, low values ($\overline{S}$ = 0.1) may discard valuable information from the magnitude of local gradients, thus jeopardizing the gradient descent search of optimal parameters. Conversely, when $\overline{S}$ = 0.3, more noise is added in the global model update (12), which complicates the training. Here, we therefore use $\overline{S}$ = 0.175.

D. Discussions

The main outcomes of the work are summarized in Table VI. In particular, using the BLSTM as a reference model, we sum up both accuracy and privacy metrics in different relevant settings, i.e., (i) fully local models (with one model by client) trained using Eqs. (3)-(4), a fully centralized model (with complete information) trained using Eqs. (3)-(4), (iii) a global model built with federated learning without DP trained using Eqs. (5)-(6), (iv) a global model built using DP-enhanced federated deep learning (DP-FDL) with Eq. (12). Models (ii), (iii) and (iv) are all trained using cross-series learning to enable cold-start forecasts for new end-users joining the coordination problem.

The complexity of the centralized model differs from the local models, i.e., the centralized model can rely on a larger architecture, and thus on higher computational capabilities since it has access to a larger database. Similarly, collaborative models in different privacy settings (fully centralized, federated learning, and DP-enhanced federated learning) are also characterized by different optimal architectures.

TABLE VI
PREDICTION PERFORMANCE AND PRIVACY LEVELS OF BLSTM-BASED PROBABILISTIC FORECASTERS TRAINED IN DIFFERENT SETTINGS.

Model	$q\|\mathcal{C}\|$	σ	r^{opt}	ϵ^{opt}	QL^{tot} [pu]
local	1	0	27	0	0.144
centralized	57	0	16	-	0.032
FL	10	0	57	-	0.054
DP-FDL	10	0.75	32	7.3	0.112
DP-FDL	10	0.25	54	167.7	0.090

We clearly observe the interest of collaborative learning since the local models achieve the worst accuracy of $QL^{\text{tot}} = 0.144$ pu. However, the improved performance from collaboration comes at the expense of privacy, by revealing sensitive input features such as raw smart meter data (exposing the periods of presence at home). The federated learning helps mitigating this issue by keeping the raw data local, while keeping a good accuracy of $QL^{\text{tot}} = 0.054$ pu. This FL approach does not provide guarantees that the trained model

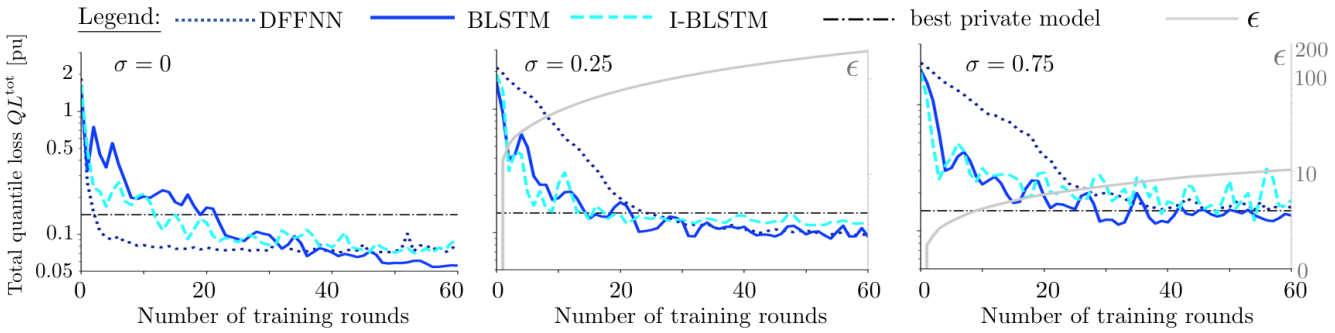

Fig. 5. Trade-off on the model performance (i.e., quantile loss) and privacy guarantees (i.e., ϵ-value) for different noise multipliers.

cannot be reversed engineered. Hence, FL is enriched with DP by introducing noise in the training. In this way, by increasing the noise multiplier σ from 0.25 to 0.75, the (worst-case) privacy breaches are significantly decreased from $\epsilon^{\text{opt}} = 167.7$ to 7.3 for a limited loss in prediction performance from 0.090 to 0.112 pu.

VI. Conclusions

This paper has presented a new privacy-preserving framework, applied to the probabilistic forecasting of nodal voltage magnitudes. The proposed framework enables to distribute the computations among the parties, and to derive a trade-off between utility and privacy by embedding the learning procedure into a differentially private mechanism. Outcomes show that compact recurrent models are inherently more robust to noise, which makes them natural candidates for the development of privacy-enhancing techniques in renewable-dominated smart grids.

As a perspective, one may be interested in tracking the privacy spent by each client, which is highly challenging since the set of clients participating in each round is private. Also, it may be useful to develop state-of-the-art attack frameworks to have an empirical evaluation of how much information an adversary can actually infer from trained models.

References

[1] J.-F. Toubeau, F. Vallée, Z. De Grève, and J. Lobry, "A new approach based on the experimental design method for the improvement of the operational efficiency in medium voltage distribution networks," *International Journal of Electrical Power & Energy Systems*, vol. 66, pp. 116–124, 2015.

[2] M. Hupez, J.-F. Toubeau, Z. De Grève, and F. Vallée, "A new cooperative framework for a fair and cost-optimal allocation of resources within a low voltage electricity community," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2201–2211, 2021.

[3] F. Moret and P. Pinson, "Energy collectives: A community and fairness based approach to future electricity markets," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3994–4004, 2019.

[4] Z. De Grève, J. Bottieau, D. Vangulick, A. Wautier, P.-D. Dapoz, A. Arrigo, J.-F. Toubeau, and F. Vallée, "Machine learning techniques for improving self-consumption in renewable energy communities," *Energies*, vol. 13, no. 18, 2020.

[5] T. Morstyn, A. Teytelboym, C. Hepburn, and M. D. McCulloch, "Integrating p2p energy trading with probabilistic distribution locational marginal pricing," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3095–3106, 2020.

[6] P. Rousseaux, J.-F. Toubeau, Z. De Grève, F. Vallée, M. Glavic, and T. Van Cutsem, "A new formulation of state estimation in distribution systems including demand and generation states," in *2015 IEEE Eindhoven PowerTech*, 2015, pp. 1–6.

[7] J. Zhao, G. Zhang, Z. Y. Dong, and M. La Scala, "Robust forecasting aided power system state estimation considering state correlations," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2658–2666, 2018.

[8] T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, and H. Zareipour, "Energy forecasting: A review and outlook," *IEEE Open Access Journal of Power and Energy*, vol. 7, pp. 376–388, 2020.

[9] A. Bracale, P. Caramia, G. Carpinelli, A. R. Di Fazio, and P. Varilone, "A bayesian-based approach for a short-term steady-state forecast of a smart grid," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 1760–1771, 2013.

[10] B. P. Hayes and M. Prodanovic, "State forecasting and operational planning for distribution network energy management systems," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 1002–1011, 2016.

[11] R. Dobbe, W. van Westering, S. Liu, D. Arnold, D. Callaway, and C. Tomlin, "Linear single- and three-phase voltage forecasting and bayesian state estimation with limited sensing," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 1674–1683, 2020.

[12] M. Hassanzadeh, C. Y. Evrenosoğlu, and L. Mili, "A short-term nodal voltage phasor forecasting method using temporal and spatial correlation," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3881–3890, 2016.

[13] M. Mokhtar, V. Robu, D. Flynn, C. Higgins, J. Whyte, C. Loughran, and F. Fulton, "Predicting the voltage distribution for low voltage networks using deep learning," in *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, 2019, pp. 1–5.

[14] A. F. Bastos, S. Santoso, V. Krishnan, and Y. Zhang, "Machine learning-based prediction of distribution network voltage and sensor allocation," in *PESGM*, 2020, pp. 1–5.

[15] Y. Wang, L. Von Krannichfeldt, T. Zufferey, and J.-F. Toubeau, "Short-term nodal voltage forecasting for power distribution grids: An ensemble learning approach," *Applied Energy*, vol. 304, p. 117880, 2021.

[16] T. Zufferey, S. Renggli, and G. Hug, "Probabilistic state forecasting and optimal voltage control in distribution grids under uncertainty," *Electric Power Systems Research*, vol. 188, p. 106562, 2020.

[17] Y. Zhang and J. Wang, "A distributed approach for wind power probabilistic forecasting considering spatio-temporal correlation without direct access to off-site information," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5714–5726, 2018.

[18] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 10–15 Jul 2018, pp. 5796–5805.

[19] C. Gonçalves, R. J. Bessa, and P. Pinson, "Privacy-preserving distributed learning for renewable energy forecasting," *IEEE Trans. Sustain. Energy*, vol. 12, no. 3, pp. 1777–1787, 2021.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, p. 1–122, 2011.

[21] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, 2017, pp. 1273–1282.

[22] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, and Q. Yang, "Secureboost: A lossless federated learning framework," 2021.

[23] Y. Wang, I. L. Bennani, X. Liu, M. Sun, and Y. Zhou, "Electricity consumer characteristics identification: A federated learning approach," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 3637–3647, 2021.

[24] N. Carlini, C. Liu, Úlfar Erlingsson, J. Kos, and D. Song, "The

secret sharer: Evaluating and testing unintended memorization in neural networks," 2019.
[25] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 3–18.
[26] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," *2019 IEEE Symposium on Security and Privacy (SP)*, May 2019.
[27] H.-Y. Tran and J. Hu, "Privacy-preserving big data analytics a comprehensive survey," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 207–218, 2019.
[28] G. Lloret-Talavera, M. Jorda, H. Servat, F. Boemer, C. Chauhan, S. Tomishima, N. N. Shah, and A. J. Pena, "Enabling homomorphically encrypted inference for large dnn models," *IEEE Transactions on Computers*, pp. 1–1, 2021.
[29] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the Third Conference on Theory of Cryptography*, ser. TCC'06. Berlin, Heidelberg: Springer-Verlag, 2006, p. 265–284.
[30] R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka, "A multi-horizon quantile recurrent forecaster," in *31st Conference on Neural Information Processing Systems (NIPS 2017), Time Series Workshop. Long Beach, CA, USA*, 2017.
[31] J.-F. Toubeau, P.-D. Dapoz, J. Bottieau, A. Wautier, Z. De Grève, and F. Vallée, "Recalibration of recurrent neural networks for short-term wind power forecasting," *Electric Power Systems Research*, vol. 190, p. 106639, 2021.
[32] J. Bottieau, L. Hubert, Z. De Grève, F. Vallée, and J.-F. Toubeau, "Very-short-term probabilistic forecasting for a risk-aware participation in the single price imbalance settlement," *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 1218–1230, 2020.
[33] R. Koenker and B. Gilbert, "Regression quantiles," *Econometrica*, vol. 46, no. 1, p. 33–50, 1978.
[34] J.-F. Toubeau, J. Bottieau, F. Vallée, and Z. De Grève, "Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1203–1215, 2019.
[35] I. Takeuchi, Q. V. Le, T. D. Sears, and A. J. Smola, "Nonparametric quantile estimation," *Journal of Machine Learning Research*, vol. 7, no. 45, pp. 1231–1264, 2006.
[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.
[37] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *International Conference on Learning Representations*, 2018.
[38] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, 2014, pp. 464–473.
[39] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015, pp. 909–910.
[40] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014. [Online]. Available: http://dx.doi.org/10.1561/0400000042
[41] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," ser. CCS, 2016.
[42] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, 2011.
[43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.
[44] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.
[45] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 4037–4049, 2017.
[46] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, 2010, pp. 51–60.
[47] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 2017, pp. 263–275.
[48] I. Mironov, K. Talwar, and L. Zhang, "Rényi differential privacy of the sampled gaussian mechanism," *CoRR*, vol. abs/1908.10530, 2019. [Online]. Available: http://arxiv.org/abs/1908.10530
[49] C. L. Canonne, G. Kamath, and T. Steinke, "The discrete gaussian for differential privacy," *CoRR*, vol. abs/2004.00010, 2020. [Online]. Available: https://arxiv.org/abs/2004.00010
[50] B. Jayaraman and D. Evans, "Evaluating differentially private machine learning in practice," in *Proceedings of the 28th USENIX Conference on Security Symposium*, ser. SEC'19. USA: USENIX Association, 2019, p. 1895–1912.
[51] N. Carlini, C. Liu, U. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *Proceedings of the 28th USENIX Conference on Security Symposium*, ser. SEC'19. USA: USENIX Association, 2019, p. 267–284.
[52] J.-F. Toubeau, J. Bottieau, Y. Wang, and F. Vallée, "Interpretable probabilistic forecasting of imbalances in renewable-dominated electricity systems," *IEEE Trans. Sustain. Energy*, vol. 13, no. 2, pp. 1267–1277, 2022.
[53] D. Freedman, R. Pisani, and R. Purves, "Statistics (international student edition)," *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

Jean-François Toubeau received the degree in civil electrical engineering, and the Ph.D. degree in electrical engineering, from the University of Mons (Belgium) in 2013 and 2018, respectively. He is currently a postdoctoral researcher of the Belgian Fund for Research (F.R.S/FNRS) within the "Power Systems and Markets Research Group" of the same University. His research mainly focuses on bridging the gap between machine learning and decision-making in modern power systems.

Fei Teng (Senior Member, IEEE) received a Ph.D. degree in electrical engineering from Imperial College London, London, U.K., in 2015. He is currently a Lecturer with the Department of Electrical and Electronic Engineering, Imperial College London, U.K. His research interests include the operation of power systems with high penetration of inverter-based resources and the security and privacy of cyber-physical systems.

Leandro Von Krannichfeldt received the B.Sc. and M.Sc. degree from the Department of Information Technology and Electrical Engineering at ETH Zurich in August 2018 and October 2020, respectively. His research interests include machine learning and optimization, especially in the field of energy forecasting.

Thomas Morstyn (Member, IEEE) received the BEng (Hon.) degree from the University of Melbourne in 2011, and the Ph.D. degree from the University of New South Wales in 2016, both in electrical engineering. He is a Lecturer of Power Electronics and Smart Grids with the School of Engineering, University of Edinburgh. His research interests include multi-agent control and market design for integrating distributed energy resources into power system operations.

Yi Wang received the B.Sc. degree from Huazhong University of Science and Technology in June 2014, and the Ph.D. degree from Tsinghua University in January 2019. He was a visiting student with the University of Washington from March 2017 to April 2018. He served as a Postdoctoral Researcher in the Power Systems Laboratory, ETH Zurich from February 2019 to August 2021. He is currently an Assistant Professor with the Department of Electrical and Electronic Engineering, The University of Hong Kong. His research interests include data analytics in smart grids, energy forecasting, multi-energy systems, Internet-of-things, cyber-physical-social energy systems.