ESIM
KU LEUVEN

Privacy-Preserving Probabilistic Forecasting in Local Energy Communities

Jean-François Toubeau, from 'Energy Systems Integration & Modeling' - ESIM

Jean-francois.toubeau@kuleuven.be

Content of the presentation

- Context:

  Why do we need forecasts in local energy communities ?

- Methodology:

  How can we provide privacy by-design forecasting models ?

  Federated learning → All data are kept local

  Horizontal (cross-series) learning → deal with users with different history

  Differential privacy → to prevent inference of raw data from the trained model

- Case study for analyzing the value of collaborative learning

- Conclusions (take home messages) & perspectives

Local energy communities

Context: The integration of distributed energy resources (DER), such as photovoltaics and electric vehicles, complicates the operation states of the power distribution grids.

Global problem: nodal voltage may violate frequently (**issues for equipment**).

Targeted solution: To proactively manage local energy exchanges, e.g., through local energy communities (LECs).

However, to ensure optimal coordination between resources, the community needs to be informed with accurate predictions of the future system state.

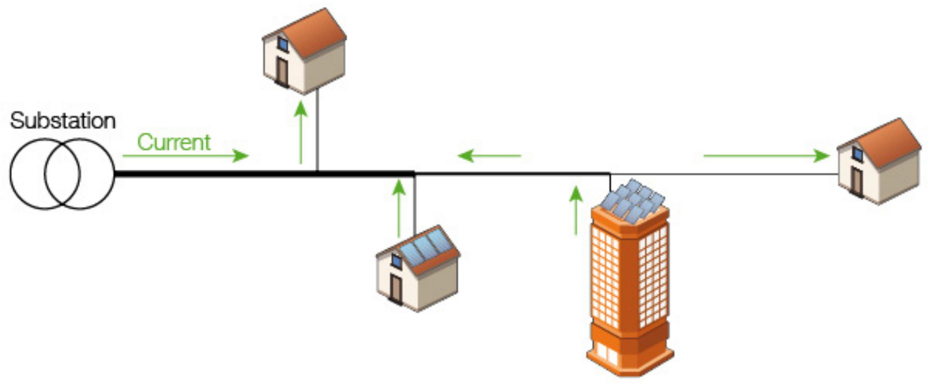
ESIM KU LEUVEN

Goal of the work

Develop a new framework for the short-term probabilistic forecasting of nodal voltage magnitudes, exploiting the information from smart metering devices.

Nodal voltages are governed by intricate:

- time correlations
- space dependencies arising from network constraints (i.e., neighboring buses are likely to exhibit similar voltage patterns).



ESIM KU LEUVEN

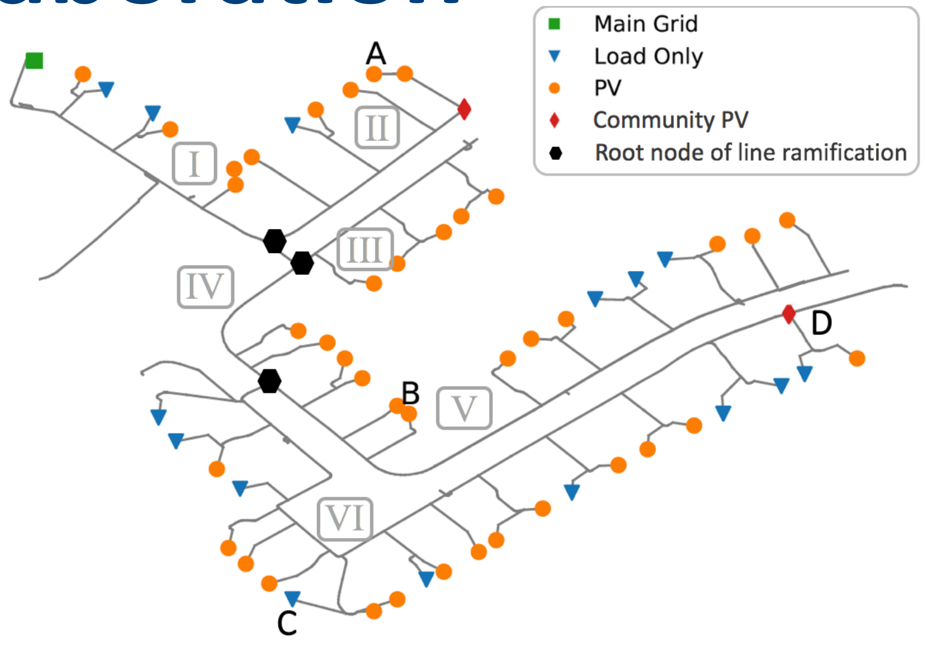
Quantifying the value of collaboration

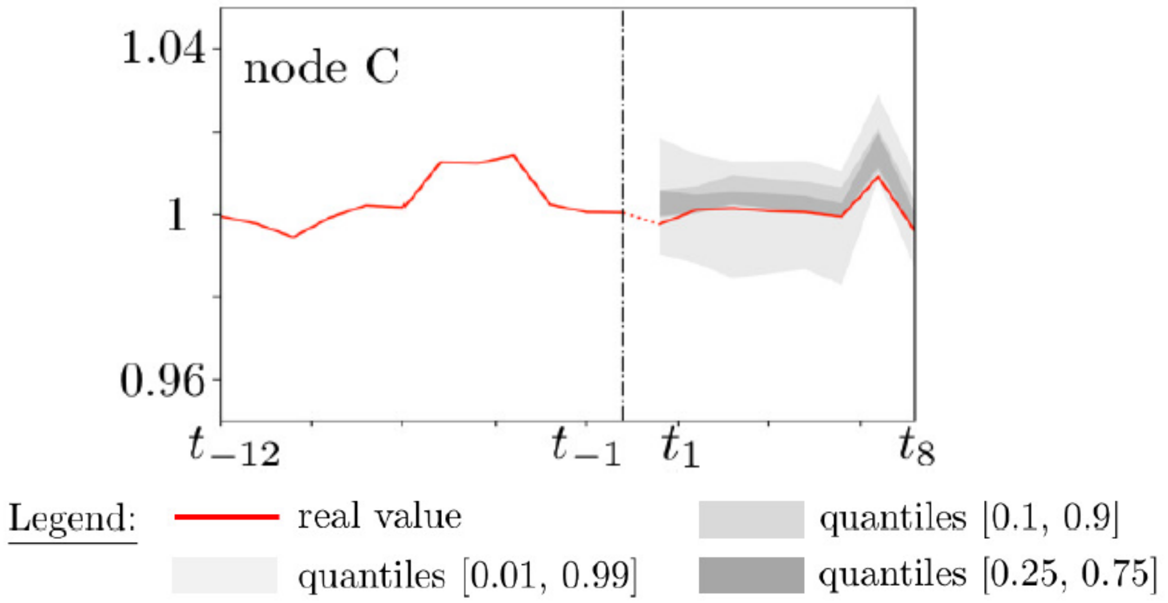
Case study:

IEEE European Low Voltage Test Feeder

Multi-horizon of 8 intervals of 30 minutes (i.e., 4 hour ahead)

A look-back window of k = 12 intervals (i.e., 6 hours) is selected to capture past dynamics





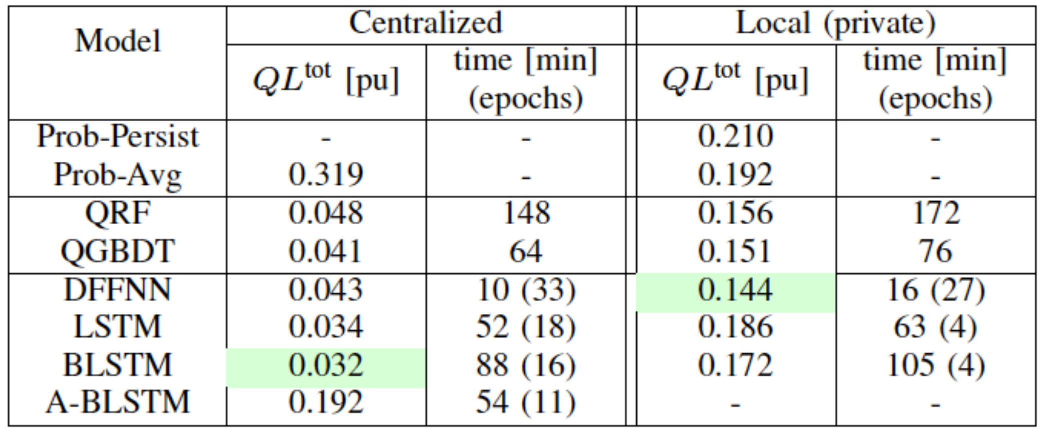
Input features:

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

Quantifying the value of collaboration

1) (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.
2) Fully private models, where a different model is trained by each end-user based only on its own private data.

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)	-	-

<u>**Quantile loss**</u> over a test set of 96 days for 57 end-users:

- High differences in performance between centralized and local models, which reflects the importance of capturing nodal dependencies between voltage levels
- RNNs fail with data scarcity

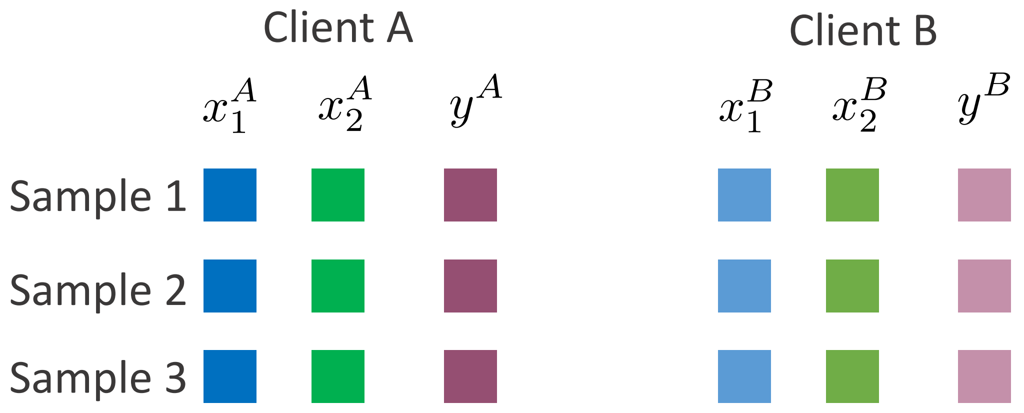
Towards privacy-preserving forecasting

Specific problem: Traditional algorithms assume that private data can be freely accessed from a centralized location.

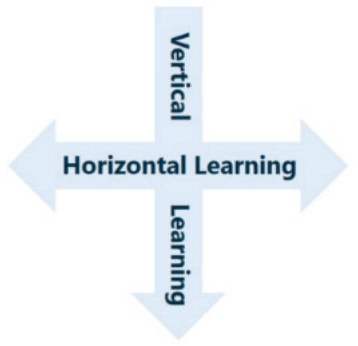
The nodal (e.g., smart meter) data are owned by end-users, who may be reluctant to share this information (as it may reveal private aspects such as home occupancy, routines, and usage of specific appliances).

➔ It is important to implement new collaborative learning strategies, leveraging the raw data of all end-users (capture dependencies among stakeholders), while complying with privacy requirements.

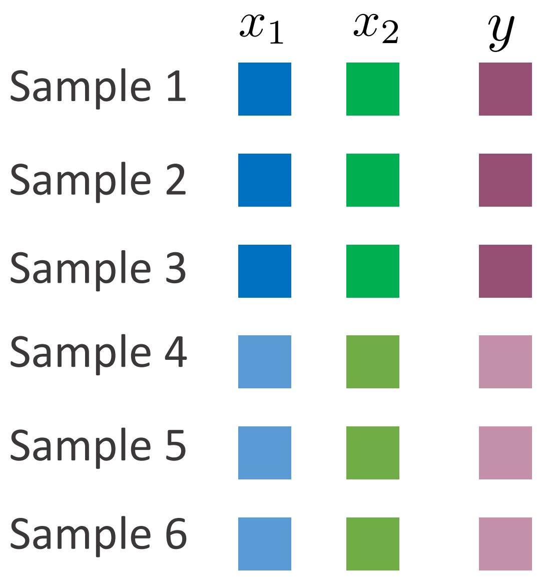
Vertical versus Horizontal learning



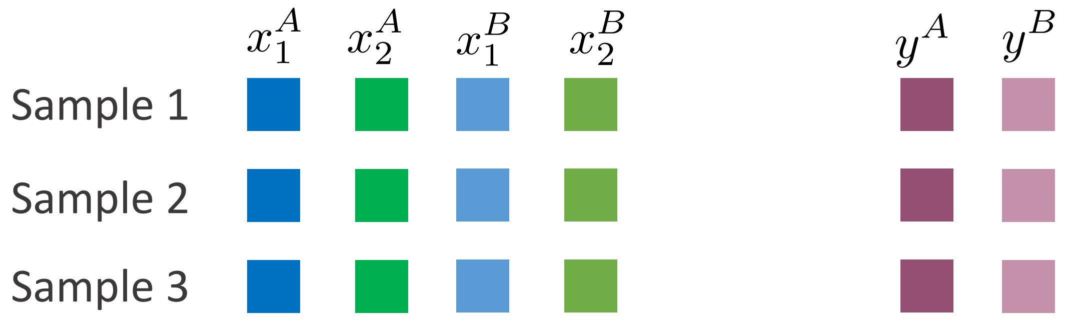
x_1 x_2	Consumption, PV generation, past voltage levels, node location, etc.
y	Voltage level



Horizontal learning

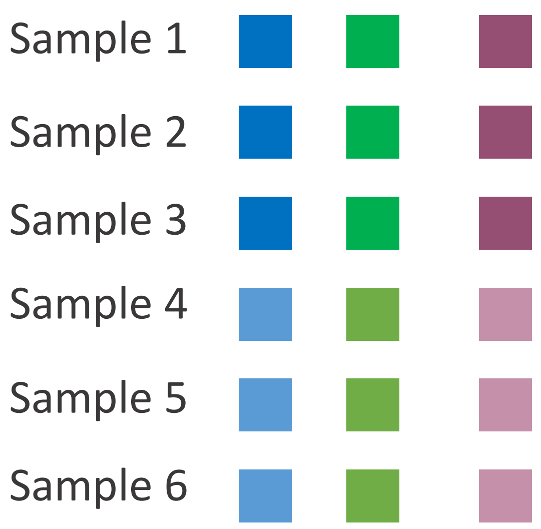


Vertical learning



Vertical versus Horizontal learning

Horizontal learning

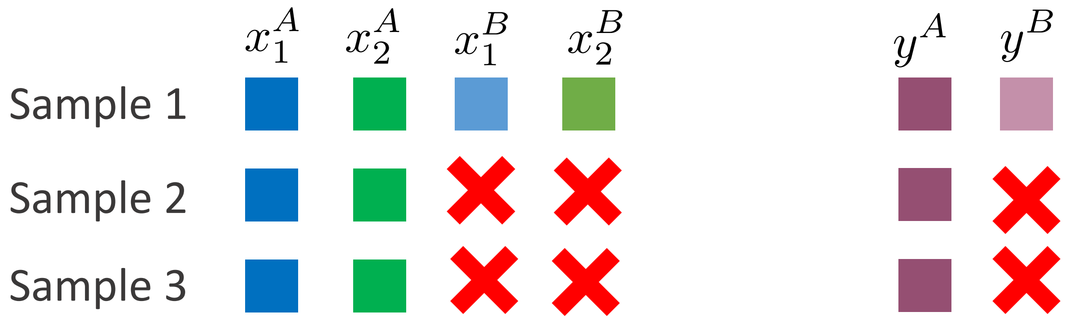


Data efficient → more samples to train the model

Can handle cold-start forecasting (for clients with very little history)

Global model → Cross-series learning for space dependencies

Vertical learning



Intuitively more suited for capturing dependencies between clients

More collaborative by design → less prone to trust issues

All clients need to have the same history (database is limited to the client with the shortest history)

Horizontal Federated Learning (FL)

Federated learning is a *distributed approach where a community of clients is coordinated by a central server to learn a global model, without sharing any raw client data.*

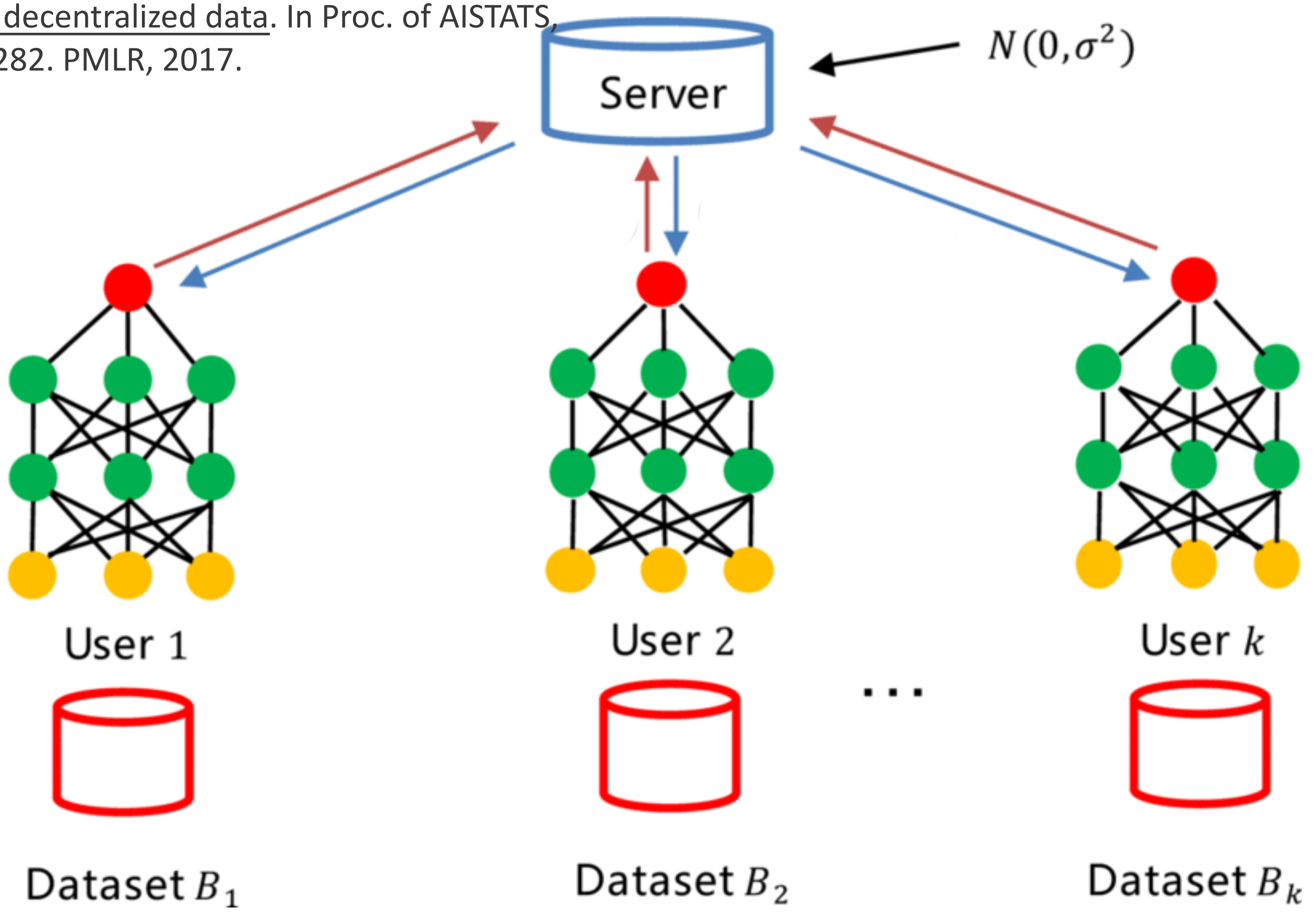
- In FL, private local data stays local, and only the trained model parameters are shared.
  - ➔ **FL provides the privacy of local data**
  - ➔ **FL avoids expensive data transfer**

- FL support an arbitrary (even dynamic) number of client nodes, which are coordinated through one central server

- FL naturally hedge against local data scarcity

Federated Averaging algorithm

[] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In Proc. of AISTATS, pages 1273– 1282. PMLR, 2017.



(1) the server initializes the parameters of the model

(2) the server broadcasts the global model to the users

(3) each selected user performs local computations on its private dataset, and computes the new (locally) optimized model

(4) The local updates are then uploaded to the server

(5) The new global model is then computed, and the procedure is iterated until convergence

Limitations of Federated Learning

- During the federated learning, the model can be accessed by adversaries to infer raw local information.
- 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

- Encrypted computations can be used to protect the training procedure, but
  - possibility to break current cryptographic functions
  - encryption schemes can be computationally expensive

Differential Privacy (DP)

- DP enables to bound and quantify the privacy leakage of sensitive information when performing learning tasks.
- DP is based on the notion of adjacent databases *D* and *D'* , which differ by the addition or removal of a single user.

(***ε***, ***δ***)-DP:

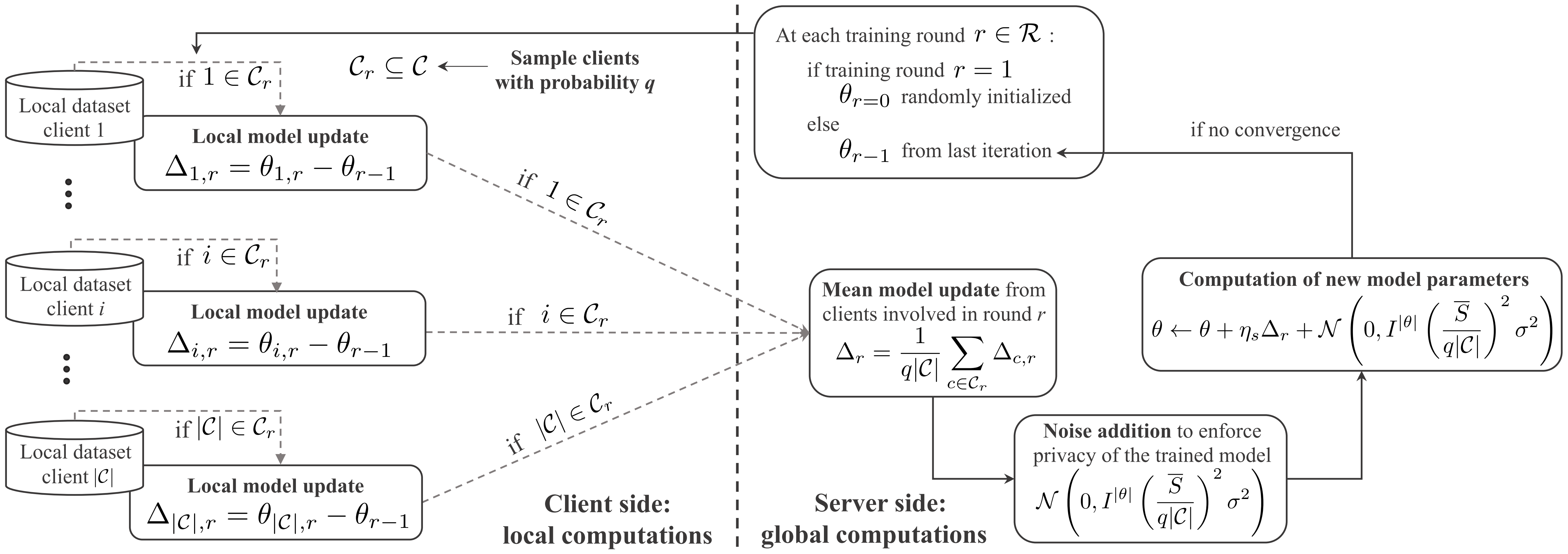
$$\Pr(\underbrace{\mathcal{M}(D)}_{=\theta_1} \in S) \leq e^{\epsilon} \Pr(\underbrace{\mathcal{M}(D')}_{=\theta_2} \in S) + \delta$$

- ***ε*** is the privacy loss yielding an upper bound of how much the probability of converging to a particular set of weights is affected by including (or removing) a single client during training
- no bound on ***ε*** with probability ***δ***

https://www.census.gov/library/video/2021/protecting-privacy-in-census-bureau-statistics.html

ESIM KU LEUVEN

Proposed framework

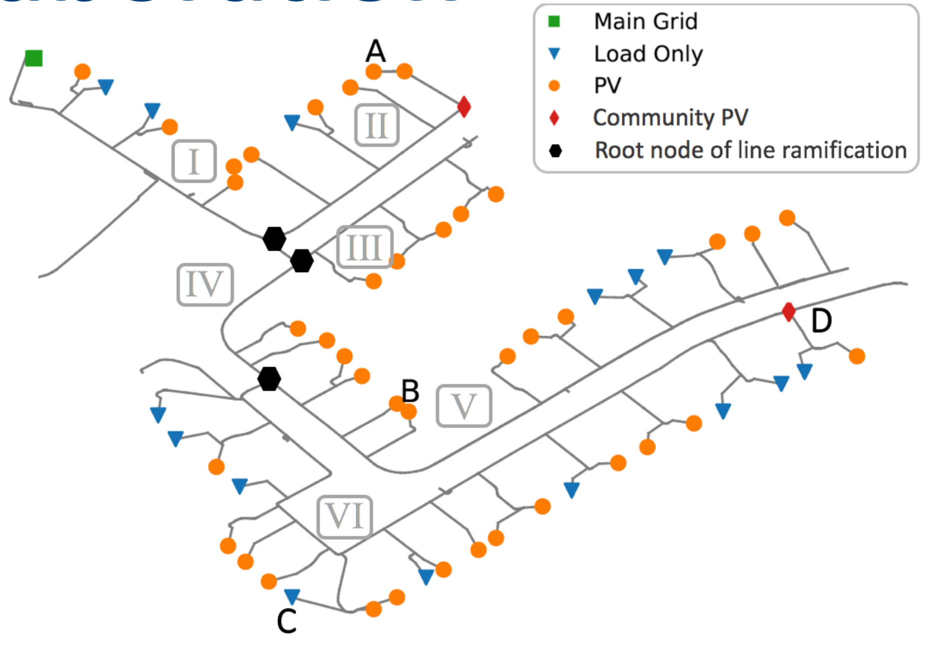


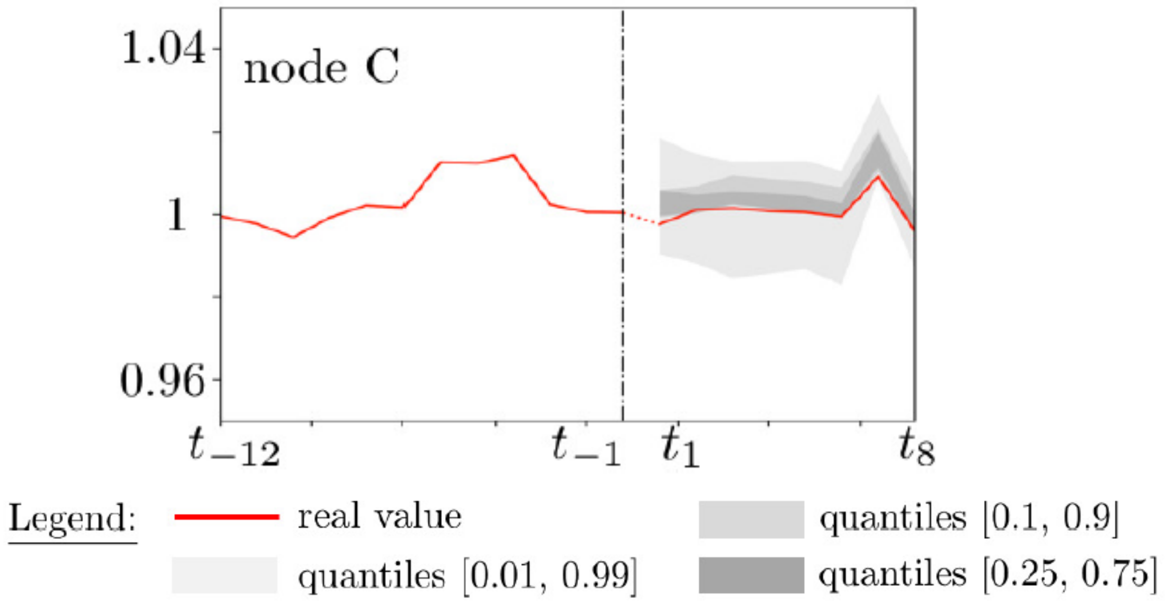
At each training round $r \in \mathcal{R}$:
if training round $r = 1$
$\theta_{r=0}$ randomly initialized
else
θ_{r-1} from last iteration
$\mathcal{C}_r \subseteq \mathcal{C}$
Sample clients with probability q
Local dataset client 1
if $1 \in \mathcal{C}_r$
Local model update
$\Delta_{1,r} = \theta_{1,r} - \theta_{r-1}$
Local dataset client i
if $i \in \mathcal{C}_r$
Local model update
$\Delta_{i,r} = \theta_{i,r} - \theta_{r-1}$
Local dataset client $|\mathcal{C}|$
if $|\mathcal{C}| \in \mathcal{C}_r$
Local model update
$\Delta_{|\mathcal{C}|,r} = \theta_{|\mathcal{C}|,r} - \theta_{r-1}$
if $1 \in \mathcal{C}_r$
if $i \in \mathcal{C}_r$
if $|\mathcal{C}| \in \mathcal{C}_r$
Mean model update from clients involved in round r
$\Delta_r = \frac{1}{q|\mathcal{C}|} \sum_{c \in \mathcal{C}_r} \Delta_{c,r}$
Noise addition to enforce privacy of the trained model
$\mathcal{N}\left(0, I^{|\theta|}\left(\frac{\overline{S}}{q|\mathcal{C}|}\right)^2 \sigma^2\right)$
Computation of new model parameters
$\theta \leftarrow \theta + \eta_s \Delta_r + \mathcal{N}\left(0, I^{|\theta|}\left(\frac{\overline{S}}{q|\mathcal{C}|}\right)^2 \sigma^2\right)$
if no convergence
Client side: local computations
Server side: global computations

ESIM KU LEUVEN

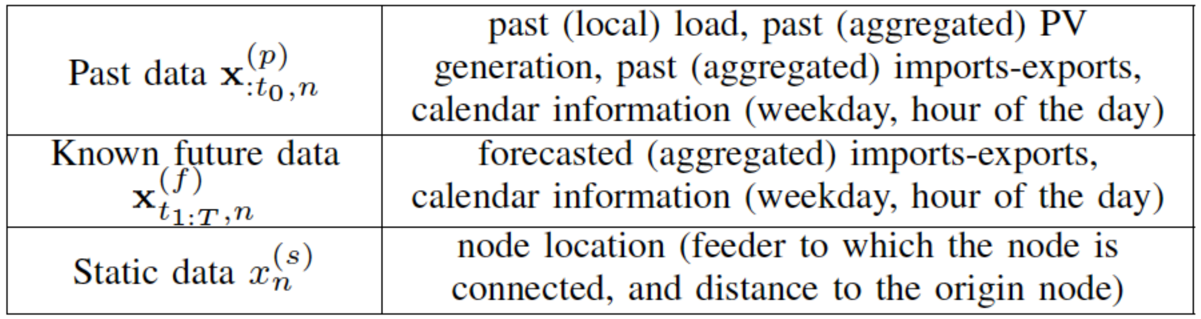
Quantifying the value of collaboration

Case study:
IEEE European Low Voltage Test Feeder
Multi-horizon of 8 intervals of 30 minutes (i.e., 4 hour ahead)
A look-back window of k = 12 intervals (i.e., 6 hours) is selected to capture past dynamics





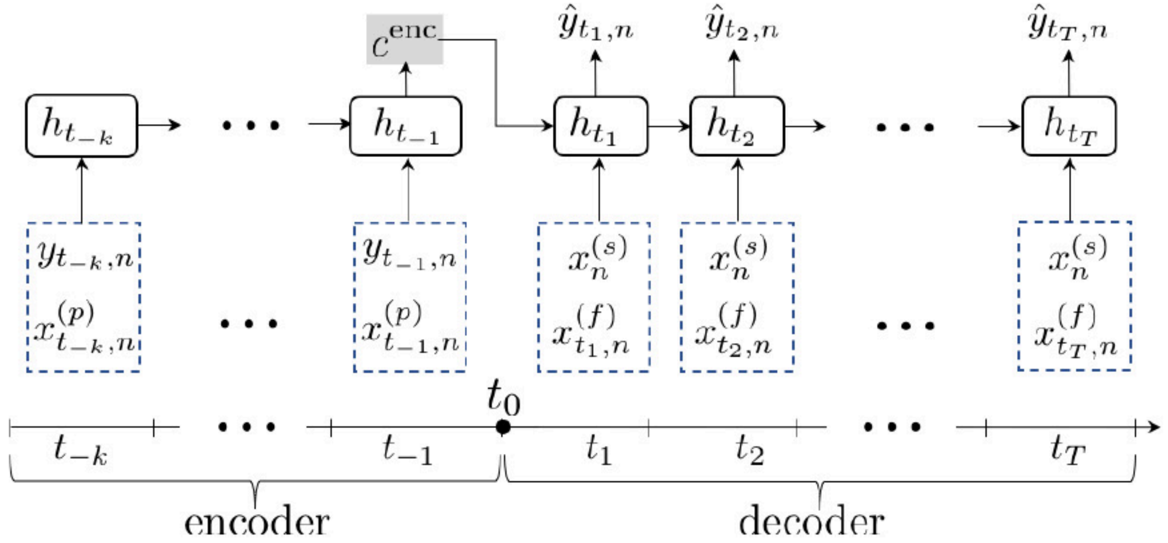
Input features:



Past data $\mathbf{x}^{(p)}_{:t_0,n}$	past (local) load, past (aggregated) PV generation, past (aggregated) imports-exports, calendar information (weekday, hour of the day)
Known future data $\mathbf{x}^{(f)}_{t_{1:T},n}$	forecasted (aggregated) imports-exports, calendar information (weekday, hour of the day)
Static data $x^{(s)}_n$	node location (feeder to which the node is connected, and distance to the origin node)

A glimpse into the RNN model

The multi-horizon time-series forecasting problem can be naturally treated as a sequence-to-sequence task.
Both encoder and decoder blocks are modeled by recurrent neural networks due to their ability to represent inter-temporal dependencies.



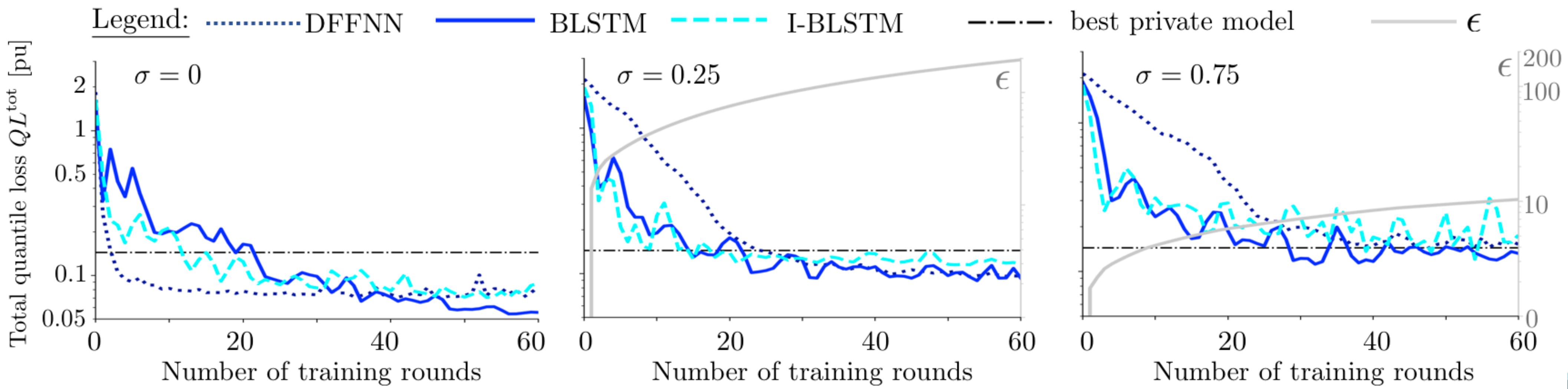
Trained with quantile regression,
for q = 1, 10, 25, 50, 75, 90 and 99 %.

➔ the output corresponding to each time step is thus 7-dimensional.

Utility privacy trade-off

There is an inherent trade-off between the performance of the prediction model and the privacy, i.e., ($\boldsymbol{\varepsilon}$, $\boldsymbol{\delta}$)-values:

- privacy breaches may occur at each round of the training
- privacy can be improved by reducing the expected number of end-users used at each round
- privacy can be controlled by the noise multiplier, which plays on the variance of the noise injected during the weight update



Main conclusions

Privacy-preserving forecasting model:

- to distribute the computations among the parties
- to derive a tradeoff 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.

J.-F. Toubeau, F. Teng, T. Morstyn, L. V. Krannichfeldt and Y. Wang, "Privacy-Preserving Probabilistic Voltage Forecasting in Local Energy Communities," in *IEEE Transactions on Smart Grid*, 2022, doi: 10.1109/TSG.2022.3187557.

ESIM KU LEUVEN

Perspectives

- Tracking the privacy spent by each client, which is highly challenging since the set of clients participating in each round is private.
- Developping state-of-the-art attack frameworks to have an empirical evaluation of how much information an adversary can actually infer from trained models.
- Using split learning, which is more tailored to perform vertical learning
- Connect with data markets, for data pricing or valuation approaches

ESIM
KU LEUVEN

Thank you for your attention

Follow ESIM on LinkedIn