Capturing Spatio-Temporal Dependencies in the Probabilistic Forecasting of Distribution Locational Marginal Prices

Jean-François Toubeau, *Member, IEEE,* Thomas Morstyn, *Member, IEEE,* Jérémie Bottieau, *Student Member, IEEE,* Kedi Zheng, *Student Member, IEEE,* Dimitra Apostolopoulou, *Member, IEEE,* Zacharie De Grève, *Member, IEEE,* Yi Wang, *Member, IEEE,* and François Vallée, *Member, IEEE*

***Abstract*—This paper presents a new spatio-temporal framework for the day-ahead probabilistic forecasting of Distribution Locational Marginal Prices (DLMPs). The approach relies on a recurrent neural network, whose architecture is enriched by introducing a deep bidirectional variant designed to capture the complex time dynamics in multi-step forecasts. In order to account for nodal price differentiation (arising from grid constraints) within a procedure that is scalable to large distribution systems, nodal DLMPs are predicted individually by a single model guided by a generic representation of the grid. This strategy offers the additional benefit to enable cold-start forecasting for new nodes with no history. Indeed, in case of topological changes, e.g. building of a new home or installation of photovoltaic panels, the forecaster intrinsically leverages the statistical information learned from neighbouring nodes to predict the new DLMP, without needing any modification of the tool. The approach is evaluated, along with several other methods, on a radial low voltage network. Outcomes highlight that relying on a compact model is a key component to boost its generalization capabilities in high-dimensionality, while indicating that the proposed tool is effective for both temporal and spatial learning.**

***Index Terms*—Electricity price forecasting, Deep learning, Bidirectional Long Short-Term Memory, Space-time correlation, Multistep-ahead time series forecasting.**

I. Introduction

WITH the advent of distributed energy resources, such as photovoltaic (PV) generation, electric vehicles and new storage technologies (e.g. home batteries), there is a growing interest in local energy markets to foster coordination between end-users [1]. In that regard, pricing energy in distribution networks, which is enabled by the massive roll-out of smart metering and energy management systems, is becoming increasingly important [2]. To account for network constraints in these local energy exchanges, an effective solution consists in penalizing energy transfers in accordance with the Distribution Locational Marginal Prices (DLMPs) [3]. Such DLMPs reflect the marginal cost of supplying an extra unit of energy at each bus (arising mainly from losses, voltage constraints and phase imbalances), thus creating nodal price differentiation when network constraints are violated. Market designs based on DLMPs have already shown some strong theoretical advantages for the future energy landscape, offering benefits not only for end-users, but also for the system as a whole [4], [5]. In particular, the implementation of DLMPs would incentivize the investment of distributed assets at optimal locations on the distribution network, while better reflecting the value of the different flexible resources.

In practice, DLMPs are the dual variables associated with the nodal energy balance constraints when solving an optimal power flow (OPF) problem that minimizes the total costs of the distribution system [6]. These DLMPs are thus complicated signals which are strongly correlated between nodes (due to technical constraints of the distribution system) and in time (along the time steps of the scheduling horizon) [7]. Additionally, these DLMPs are highly uncertain, since the OPF problem is subject to different stochastic sources, i.e. upstream energy prices, as well as local PV generation and consumption that have a significant impact on power flows.

Our objective is thus to develop a data-driven tool providing reliable probabilistic prediction of DLMPs that will be used by distribution system operators (DSOs) to properly motivate end-users to contribute to the network support during their subsequent trading process [8]. In particular, we aim at developing a prediction framework that bypasses the need for the DSO of solving a day-ahead multi-phase distribution system dispatch, in a probabilistic environment that accurately represents correlations among the many uncertainty sources. This aspect is of high interest since determining the economic dispatch requires the DSO to gather the preferences and resource characteristics of all the consumers, which would necessitate significant bidirectional communication and may have privacy implications. Overall, the proposed data-driven tool is designed to be applicable to any low-voltage (LV) system, for predicting (in day-ahead) the DLMPs embodying the intricate space-time correlations, while quantifying the uncertainty related to future conditions.

This task requires that each position of the space-time graph

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 post-doctoral research fellow of the National Fund of Scientific Research (FNRS) at the Electrical Power Engineering Unit, University of Mons, Belgium.

J. Bottieau, Z. De Grève and F. Vallée are with the same unit.

T. Morstyn works in the school of Engineering, University of Edinburgh, Edinburgh EH9 3JL, U.K.

K. Zheng is with the State Key Laboratory of Power Systems,Department of Electrical Engineering, Tsinghua University, Beijing 100084, China.

D. Apostolopoulou is a lecturer at the Department of Electrical and Electronic Engineering at City, University of London.

Y. Wang is with Power Systems Laboratory, Department of Information Technology and Electrical Engineering, ETH Zürich, Switzerland.

(i.e. each node of the distribution feeder for each time step) has access to information from all positions [9]-[11], which is difficult to achieve in a compact and robust framework. In that respect, traditional machine learning techniques, such as random forests, support vector machines and feedforward neural networks, are designed for a static learning of the relationship between the variables of interest (i.e. outputs) and their covariates (i.e. inputs) [12]. Such models are thus known to struggle at efficiently sharing information among different space-time locations [13].

A naive approach to account for such dependencies consists thus in splitting the complexity of the task by relying on multiple models, e.g. a different model is trained for each point of the space-time domain [14]. However, such a procedure does not scale well to large systems since the number of models to train (and store) increases with the problem dimensionality. Moreover, it necessitates a cumbersome (engineering-based) data pre-processing to feed each model with the relevant neighbouring information [15]. Another solution consists in simultaneously predicting all outputs of the forecast domain (in a single instance) [16]. However, when the number of outputs increases (with many clients over a long forecast horizon), this architecture typically leads to optimization difficulties to efficiently map the resulting high-dimensional input features to the high-dimensional output vector [17]. This issue is further exacerbated when few relevant historical data are available.

In order to improve such strategies, statistical methods (such as autoregressive models) have been developed with the goal of processing and learning sequences where the elements are strongly correlated over time [18]. Different alternatives have then been proposed to integrate spatial information into such models. In [19], [20], the spatial dimension is represented through a vector autoregressive (VAR) model, where co-dependence between sites is captured by additional coefficients. In [21], the interdependence structure among locations and lead times is modeled with multivariate ellipsoids. In such models, the correlations are imposed *a priori*, which is not suited for representing the varying correlation pattern of DLMPs. Indeed, nodal dependencies strongly differs between safe operation conditions where all nodal prices are equal, and stressed situations where price discrepancies arise. Moreover, the linear nature of the model leads to limitations in the ability to represent nonlinear dependencies and high-frequency events (such as rapid variations between successive time steps) [22].

To better represent complex dependencies, one can rely on recurrent neural networks (RNNs), which have recently achieved improved performance in many tasks such as the short-term prediction of electrical series (load, renewable generation and prices) [23]-[26]. In particular, their emergence has been fostered by the Long Short-Term Memory (LSTM) architecture, which is characterized by a memory cell that is able to extend the range of temporal context available to the model [27]. This basic (one-dimensional) architecture has been generalized towards the space-time domain in [28], by introducing convolutional LSTM (ConvLSTM) for the task of precipitation forecasting. However, such an architecture requires that the data follows a matrix structure, where the spatial information is divided into (2-dimensional) equal subspaces which does not suit the radial topology of low-voltage systems. In [29], a LSTM-based network is used to extract relevant features of different (spatially-correlated) wind farms, which are then fed into a deterministic prediction model combining graph theory and convolutional neural networks.

Overall, applying these approaches for DLMPs forecasting involves tailoring the model to a specific architecture of the low-voltage system, such that the resulting model is not robust in case of topological modification. In particular, accommodating new nodes (e.g. construction of a new home, installation of community PV, etc.) requires to modify the forecaster architecture, and to resort to reliable assumptions to infer the historical missing data, which may not be trivial.

In this paper, we develop a generic model that is able to cope with new clients (with no history) in a framework that exploits space-time dependencies. The contributions are threefold.

Firstly, we use the flexible nature of neural networks to represent the high-level spatio-temporal structure of DLMPs. Practically, the tool relies on a deep bidirectional LSTM network, which is designed to share the information among all time steps of the prediction horizon. We find that this structure yields a large improvement over the standard LSTM, thus showing a great potential for other multi-step forecast applications. This solution is combined with a new generic method to encode spatial information within the model inputs, which includes both the nodal position indication and the grid structure. These data enrich the model with the ability to account for nodal price differentiation.

Secondly, in order to stimulate cross-nodal learning within a procedure that is applicable to any distribution system, each nodal DLMP is considered as a different sample fed into a single model. This solution boosts the model generalization capabilities, while inherently supporting cold-start forecasting for new nodes with no historical values [30]. Indeed, for any new client, the procedure only requires to encode its spatial information, and we can use the model (which is trained with the past information of other nodes) to obtain the desired predictions, without needing to retrain the tool from scratch.

Thirdly, an extensive comparison with other state-of-the-art forecasting approaches in a probabilistic setting is carried out. The benchmark intends not only to compare most successful tools such as gradient boosting and deep feedforward neural networks, but also different strategies to represent space-time correlations. Outcomes show that avoiding to deal with high-dimensionality in both input and output feature spaces is essential to obtain an efficient model, and that the proposed tool is an efficient architecture to compactly leverage space-time information, thereby outperforming other forecasters.

Overall, the resulting model is thus scalable in time (through a bidirectional recurrent model) and in space (since the model predicts each nodal DLMP individually). Moreover, the generic and data-driven nature of the model makes it ideally suited for smart grid applications where a different model can be efficiently applied to each of the many low-voltage areas.

The proposed deep bidirectional LSTM model is presented in detail in Section 2, together with the strategy to encode spatial information. Section 3 defines the different modeling frameworks used as benchmark to capture space-time depen-

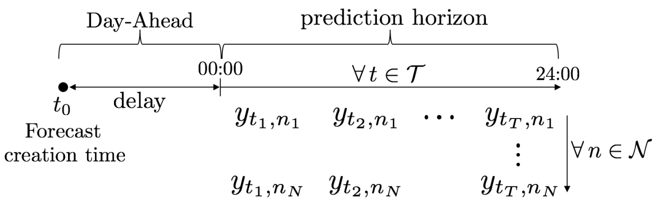
Day-Ahead
prediction horizon
00:00
$\forall t \in \mathcal{T}$
24:00
t_0
delay
Forecast creation time
y_{t_1,n_1} y_{t_2,n_1} $\cdots$ y_{t_T,n_1}
y_{t_1,n_N} y_{t_2,n_N} y_{t_T,n_N}
$\forall n \in \mathcal{N}$

Fig. 1. Representation of the day-ahead forecasting problem, i.e. at the forecast creation time t_0, the model jointly provides the predicted DLMPs $y_{t,n}$ for every grid nodes $n \in \mathcal{N} = \{n_1, \dots, n_N\}$ for all time periods $t \in \mathcal{T} = \{t_1, ..., t_T\}$ of the next day.

dencies in the prediction of DLMPs. In Section 4, these models are evaluated on a 57-buses low-voltage distribution system with a complex tree-structured topology. Finally, Section 5 concludes the paper and outlines the main results.

II. Model description

The objective is to generate (in day-ahead) reliable probabilistic forecasts of DLMPs, so as to properly inform local energy exchanges, and thereby supporting an optimized management of the low-voltage (LV) system [3].

As represented in Fig. 1, there is a delay (of typically 12 hours) between the forecast creation time t_0 and the start of the prediction horizon t_1, which differentiates this problem from traditional online prediction tasks. Overall, the forecaster is designed to solve the following time series regression problem:

$$p\left(y_{t_1,n}, ..., y_{t_T,n} \middle| y_{:t_0,n}, x^{(f)}_{t_0:,n}, x^{(s)}_n\right) \forall n \in \mathcal{N} \tag{1}$$

where the goal is to forecast DLMPs for each grid node $n \in \mathcal{N} = \{n_1, ..., n_N\}$ over the $t \in \mathcal{T}$ steps (from t_1 to t_T) of the daily horizon. To that end, the tool has access to different explanatory variables (inputs), i.e. the DLMP values $y_{:t_0,n}$ known at the forecast creation time t_0 as well as the temporal covariates $x^{(f)}_{t_1:,n}$ (such as the estimated consumption and PV generation), and the static features $x^{(s)}_n$ (such as the node location features) that do not vary with time.

This task is difficult since electricity prices are non-stationary signals that show multiple periodicities (both slow and fast fluctuating components) with strong space-time correlations [31]. To that end, based on [25], we propose in Section II-A a modeling framework that uses the ability of bidirectional LSTM recurrent neural networks to access long-range context along time dimension. Then, in Section II-B, the procedure is complemented to foster cross-nodal learning (between nodes of the distribution system). This is achieved by encoding spatial information as additional explanatory variables, while feeding each nodal price series as a different sample into a single model. Finally, the input selection and training framework are described in Section II-C.

A. Capturing time dependencies

Recurrent neural networks (RNNs) are models designed to process input series through the recursive application of a transition function $\mathcal{H}$ at each time period. Such networks are characterized by a time-dependent hidden state h_t that provides an internal representation of past events, which is used to propagate relevant information through time (Fig. 2). Indeed, at each time step $t \in [t_1, t_T]$, the transition function $\mathcal{H}$ maps the hidden state h_t to both local features (inputs) x_t and the previous hidden state h_{t-1}.

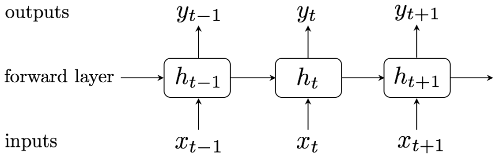
outputs
forward layer
inputs
y_{t-1} y_t y_{t+1}
h_{t-1} h_t h_{t+1}
x_{t-1} x_t x_{t+1}

Fig. 2. Recurrent neural network.

Basic RNN architectures have shown a limited ability in grasping dependencies more than a few time steps long [27]. This problem is alleviated by the introduction of the Long Short-Term Memory (LSTM) transition function, which is characterized by an additional hidden state c_t designed to act as a memory for keeping long-term information from past inputs (5). This memory cell c_t interacts with three control gates, i.e. the input gate i_t which memorizes the new information revealed over time (2), the forget gate f_t which has the ability to discard irrelevant information from the past (3), and the output gate o_t that extracts the relevant information from the memory content c_t to compute the LSTM state h_t (4). Since the neural network is composed of multiple LSTM neurons, the information can be either propagated or eliminated among different units such that the tool is potentially able to model any complex nonlinear signals, resulting in performance enhancement [32]. The standard LSTM is implemented by the following composite function $\mathcal{H}_{LSTM}$:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right), \tag{2}$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right), \tag{3}$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right), \tag{4}$$

$$c_t = i_t \odot \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) + f_t \odot c_{t-1}, \tag{5}$$

$$h_t = o_t \odot \tanh(c_t), \tag{6}$$

where $W.$ and $U.$ are the weight matrices, while $b.$ are the bias vectors, all of which representing the parameters of the neural network (that need to be optimized during the training phase to efficiently predict DLMPs). Also, $\odot$ denotes element-wise multiplication, and σ is the logistic sigmoid function.

As depicted in Fig. 2, standard LSTM networks process inputs in temporal order, i.e. the DLMP at t_k is predicted using only data from $[t_1, t_k]$ such that the predictions are only based on previous context. This framework is perfectly suited for online tasks, but may be plagued with a loss of valuable information for our day-ahead problem. Indeed, since the whole DLMP sequence $y_{t_1,n}, ..., y_{t_T,n}$ needs to be jointly predicted at once (Fig. 1), there is no reason not to exploit the available context in $[t_{k+1}, t_T]$. The underlying idea is that the available knowledge related to time t_{k+i} with $i = 1, \dots, T-k$, such as the estimated values of aggregated loads and PV generation at the LV system level, can help at explaining the price conditions at time t_k. This logic has been successfully applied in translation tasks where sentences that seem meaningless after a few words are found to become intelligible in the light of future context [33].

A powerful strategy to capture such backward dependencies is to rely on a bidirectional LSTM (BLSTM). This tool is built upon two separate hidden layers $\mathcal{H}_{LSTM}$ (2)-(6), each one processing the data in opposite directions. The BLSTM model (Fig. 3) computes the forward hidden sequence $\overrightarrow{h}_t$ by iterating from $t = 1$ to T, and the backward hidden sequence $\overleftarrow{h}_t$ by iterating from $t = T$ to 1. These vectors are then fed into the same output layer to generate the DLMP predictions y_t (9). Hence, for every point t of the sequence, the BLSTM has complete information about all points before and after t.

$$\overrightarrow{h}_t = \mathcal{H}_{LSTM}\left(W_{\overrightarrow{h}} x_t + U_{\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right), \tag{7}$$

$$\overleftarrow{h}_t = \mathcal{H}_{LSTM}\left(W_{\overleftarrow{h}} x_t + U_{\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right), \tag{8}$$

$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y. \tag{9}$$

Finally, we further strengthen the tool by using deep architectures to extract more information from input features [34]. Deep RNNs are created by stacking multiple RNN hidden layers on top of each other, with the output h^l of one layer l forming the input for the next $l + 1$. When combining deep architectures with the bidirectional data processing, each hidden state h^l is replaced by the concatenation of forward and backward states $\overrightarrow{h}^l$ and $\overleftarrow{h}^l$. Practically, for layers $l > 1$, the input x_t in (7)-(8) is replaced by the concatenation of the outputs $\left(\overrightarrow{h}_t^{l-1}, \overleftarrow{h}_t^{l-1}\right)$ of the bidirectional layers at the level $l - 1$ below. The prediction y_t is computed in (9) using the hidden vectors $\overrightarrow{h}_t^L$ and $\overleftarrow{h}_t^L$ of the upper layer L.

B. Capturing nodal dependencies

In its standard form, the deep BLSTM network strictly focuses on processing sequential data. In order to capture space dependencies (i.e. nodal price differentiation due to grid constraints), we train a single model in a framework where each nodal price series $y_{\cdot,n}$ is individually forecasted.

This strategy offers four advantages. Firstly, training a single model on multiple nodal series allows the sharing of statistical information across nodes, thereby triggering cross-series learning. Secondly, the training dataset is $|\mathcal{N}|$ times larger than models where nodes are jointly predicted, so that overfitting risk reduces. Thirdly, the framework efficiently decouples the size of the distribution system from the dimensionality of the output space $y_{t,n}$ treated by the model. This prevents scalability issues associated with approaches jointly predicting all nodal prices $\{y_{t,n_1}, ..., y_{t,n_N}\}$ in a single instance. Fourtly, since the model is intrinsically trained to generalize to all nodes $n \in \mathcal{N}$ of the distribution system, it has the ability to generate cold-start predictions for nodes with little or no history (such as new homes). In order for the tool to exploit this ability, we need to properly express spatial data.

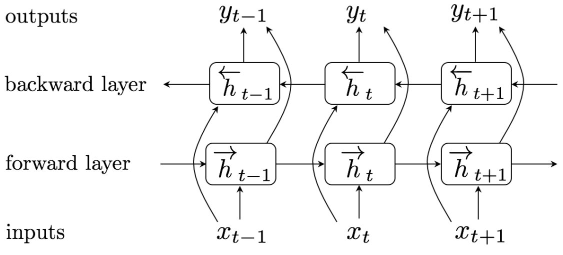

Fig. 3. Bidirectional recurrent neural network.

These spatial features must represent the location of the nodes, while accounting for the structure of the distribution system. Traditional methods rely on discrete variables, but such strategies face the issue that node n_2 is not 2 times more important than n_1. Moreover, they are unable to capture the similarity between nodes of concomitant branches in complex tree-structured systems. In this work, we therefore use a binary representation, which offers a more generic way of representing spatial data. Practically, each branch of the system is associated with a boolean feature, which is equal to 1 for the nodes connected to the branch, and 0 for the others. This information is then complemented by encoding the distance between each client and the root node of the branch.

This generic framework separates the size of the distribution system from the dimension of the input-output space, thus accommodating new nodes without affecting the structure of the forecaster. Indeed, in case of a new connection, we only need to encode its spatial information, and the forecaster leverages its generalization capabilities learned from past observations of other clients to generate the DLMP predictions of interest.

C. Inputs selection and model training

In addition to spatial features (described in Section II-B), the input vector x must be enriched with relevant explanatory variables. Since DLMPs are mainly dependent on global loading conditions within the system, the forecaster is guided by the aggregated conditions at the low-voltage substation level. In particular, the model takes as inputs the forecasted global PV generation and load consumption. Moreover, calendar-based features, i.e. hour of the day, and day of the week, are also represented with a binary representation [25]. In this work, no historical prices (such as previous day, or previous week) are used as explanatory variables in neural networks.

In general, the use of neural networks is divided into two stages. Firstly, in the training phase, we have access to the historical database, and the objective is to optimize the parameters θ of the forecaster (corresponding to the weight and bias matrices for neural networks) such that we accurately map the output y corresponding to a given input x. Secondly, once the model is trained, it can be used for actual field forecasting.

During the training, the optimal model parameters θ^* are determined by minimizing a loss function $\mathcal{L}$ between the actual values y and the predictions $\hat{y}$:

$$\theta^* = \arg\min_{\theta} \mathcal{L}\big(\hat{y}(x, \theta), y\big) \tag{10}$$

The loss function $\mathcal{L}$ is a user-defined measure that quantifies how well the model fits the historical data. In traditional regression models, the goal is to minimize the mean squared error $\sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \left(\hat{y}_{t,n} - y_{t,n}\right)^2$, which yields a deterministic forecast reflecting the conditional mean $\mathbb{E}\left(y_{t,n} \mid x_{t,n}\right)$ where $x_{t,n}$ denotes all explanatory variables known at t_0.

In order to properly consider the uncertainty around predictions, two distinct philosophies can be found. Firstly, one can implement a two-step procedure whereby a point forecasting

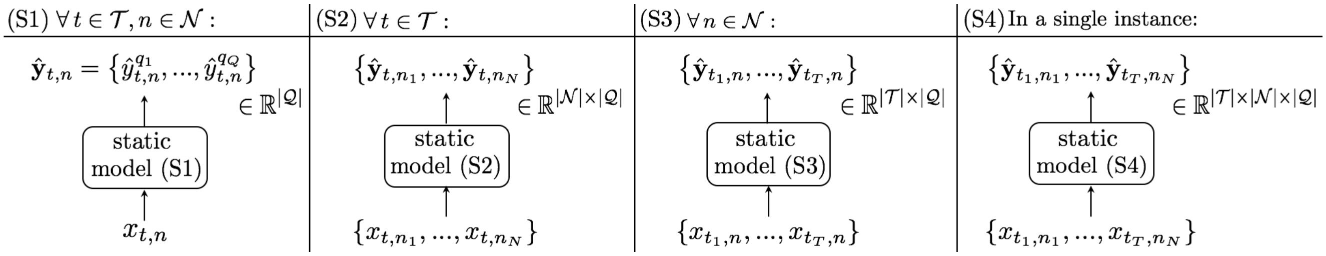
(S1) $\forall\, t \in \mathcal{T}, n \in \mathcal{N}$:
$\hat{\mathbf{y}}_{t,n} = \{\hat{y}^{q_1}_{t,n}, ..., \hat{y}^{q_Q}_{t,n}\}$
$\in \mathbb{R}^{|\mathcal{Q}|}$
static model (S1)
$x_{t,n}$
(S2) $\forall\, t \in \mathcal{T}$:
$\{\hat{\mathbf{y}}_{t,n_1}, ..., \hat{\mathbf{y}}_{t,n_N}\}$
$\in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{Q}|}$
static model (S2)
$\{x_{t,n_1}, ..., x_{t,n_N}\}$
(S3) $\forall\, n \in \mathcal{N}$:
$\{\hat{\mathbf{y}}_{t_1,n}, ..., \hat{\mathbf{y}}_{t_T,n}\}$
$\in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{Q}|}$
static model (S3)
$\{x_{t_1,n}, ..., x_{t_T,n}\}$
(S4) In a single instance:
$\{\hat{\mathbf{y}}_{t_1,n_1}, ..., \hat{\mathbf{y}}_{t_T,n_N}\}$
$\in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{N}|\times|\mathcal{Q}|}$
static model (S4)
$\{x_{t_1,n_1}, ..., x_{t_T,n_N}\}$

Fig. 4. Strategies to represent space-time dependencies with static models, (S1) probabilistic DLMPs are individually predicted for each time step $t \in \mathcal{T}$ at each node $n \in \mathcal{N}$, (S2) the model is run for each time step individually, and all nodal prices are jointly predicted, (S3) the model is run for each node individually, and all time periods are jointly predicted, (S4) all DLMPs of the space-time horizon are predicted together.

is firstly obtained and a distribution should then be estimated to calibrate the point results and get a final density forecast. In particular, an efficient framework to approximate the posterior distribution that quantifies the prediction uncertainty consists in using variational inference [35]-[37]. Secondly, there are methods directly providing the probabilistic predictions. This can be achieved using either a fully parametrized model (assuming, e.g., a Gaussian distribution of the error) or via an empirical function. In this paper, the latter approach is selected, where a tailored quantile regression tool is developed.

Practically, the model is trained with the goal of predicting the conditional quantiles $\hat{y}^q_{t,n}|x_{t,n}$ for different $q \in \mathcal{Q} \in [0,1]$, within a non-parametric (distribution free) method. The resulting $|\mathcal{Q}|$-dimensional output $\hat{\mathbf{y}}_{t,n} = \{\hat{y}^{q_1}_{t,n}, ..., \hat{y}^{q_Q}_{t,n}\}$ is forecasted within a single compact network.

To that end, we minimize the pinball loss $\mathcal{L}^q$, which yields a trade-off between calibration and sharpness [38]. A calibrated model ensures the statistical correctness of the predictions, i.e. the percentage of values $y_{t,n}$ (across all t and n) below the predicted quantile $\hat{y}^q_{t,n}$ is close to the nominal probability q. The sharpness ensures that the prediction interval widths (between quantiles) are sufficiently narrow to provide useful information.

$$\mathcal{L}^q(y,\hat{y}) = q \max(y-\hat{y},0) + (1-q)\max(\hat{y}-y,0) \qquad (11)$$

However, the standard pinball loss is not differentiable when the forecast error is zero, i.e. $\hat{y}^q_{t,n} = y_{t,n}$, which prevents the use of gradient descent-based methods to train the model. The loss function is thus smoothly approximated by including the Huber norm [39], which consists in replacing the pinball function by the (continuously differentiable) Euclidean norm when the error is lower than a user-defined threshold ϵ (in this paper, we arbitrarily use $\epsilon = 10^{-6}$):

$$\mathcal{L}^q_{H_b}(y,\hat{y}) = q \max(H_b(y,\hat{y}),0) + (1-q)\max(H_b(y,\hat{y}),0) \qquad (12)$$

where the Huber norm $H_b\,(y,\hat{y})$ is computed as:

$$H_b\,(y,\hat{y}) = \begin{cases} \frac{(\hat{y}-y)^2}{2\epsilon} & 0 \le |\hat{y}-y| \le \epsilon \\ |\hat{y}-y| - \frac{\epsilon}{2} & |\hat{y}-y| > \epsilon \end{cases} \qquad (13)$$

In this work, the model is trained to minimize the total loss $\sum_{t\in\mathcal{T}} \sum_{n\in\mathcal{N}} \sum_{q\in\mathcal{Q}} \mathcal{L}^q_{H_b}(y_{t,n}, \hat{y}^q_{t,n})$, using the Adam algorithm, a stochastic gradient descent method that uses adaptive learning rates for escaping local optima.

III. Benchmarks

In this section, we introduce different probabilistic forecasting methods as benchmarks for the case study.

A. Static models

In this part, we implement four different static models, i.e. which do not endogenously represent dependencies between points of the space-time horizon. These are shown in Fig. 4.

The static model of reference (S1) consists in individually forecasting each point (t,n) of the space-time domain, i.e. the model is run $|\mathcal{T}|$*$|\mathcal{N}|$ times to generate a $|\mathcal{Q}|$-dimensional output $\hat{\mathbf{y}}_{t,n} = \{\hat{y}^{q_1}_{t,n}, ..., \hat{y}^{q_Q}_{t,n}\}$ at each iteration.

The second topology (S2) aims at jointly forecasting all nodes at a given time period, i.e. the model is run $|\mathcal{T}|$ times to generate the $|\mathcal{N}|$*$|\mathcal{Q}|$-dimensional output.

The third architecture (S3) is trained to jointly predict all times periods for each particular node, i.e. the model is applied $|\mathcal{N}|$ times to provide the $|\mathcal{T}|$*$|\mathcal{Q}|$-dimensional output.

Finally, the fourth strategy (S4) is designed to forecast all points of the spatio-temporal horizon in a single instance, i.e. with an output of size $|\mathcal{T}|$*$|\mathcal{N}|$*$|\mathcal{Q}|$.

It is important to notice that, since all nodes are jointly predicted in models (S2) and (S4), it is irrelevant to feed them with spatial information (Section II-B). These four strategies (S1)-(S4) are applied for the four forecasting tools described hereunder, resulting in 4*4 = 16 tested cases.

- *FFNN*, a feedforward neural network characterized by a single hidden layer composed of neurons with rectifier linear units (ReLUs) as activation function. This tool is the basic neural network structure, which is theoretically able to learn any nonlinear function.
- *DFFNN*, a deep feedforward neural network, composed of several hidden layers stacked on top of each other with the goal of building up higher level representations of data, which enables to more efficiently map the raw available inputs to the desired predictions.
- *QRF*, a quantile regression forest, i.e. a method that generalizes random forests for estimating quantiles instead of the conditional mean. The number of trees is set to 500, which makes the QRF a strong learning model.
- *GradBoost*, a gradient boosting regression tree trained with the pinball loss to generate quantile predictions. This method sequentially creates new models to forecast the residuals of the global model obtained at the previous stage. The number of boosting stages is fixed to 100.

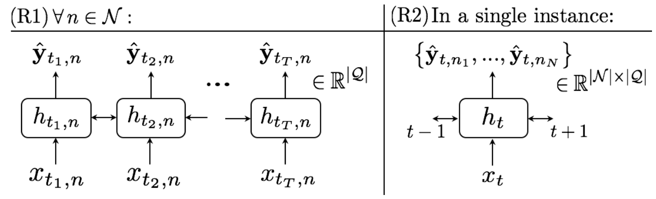
(R1) ∀n ∈ 𝒩 :
$\hat{\mathbf{y}}_{t_1,n}$
$\hat{\mathbf{y}}_{t_2,n}$
$\hat{\mathbf{y}}_{t_T,n}$
$\in \mathbb{R}^{|\mathcal{Q}|}$
$h_{t_1,n}$
$h_{t_2,n}$
$h_{t_T,n}$
$x_{t_1,n}$
$x_{t_2,n}$
$x_{t_T,n}$
(R2) In a single instance:
$\{\hat{\mathbf{y}}_{t,n_1}, ..., \hat{\mathbf{y}}_{t,n_N}\}$
$\in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{Q}|}$
$t-1$
h_t
$t+1$
x_t

Fig. 5. Strategies to represent space-time dependencies with time-dependent models, (R1) by encoding spatial information so as to predict each node individually (in a more compact model enabling cold-start forecasting), and (R2) by jointly predicting all nodes of the distribution system.

B. Recurrent models

We focus here on time-dependent models. As represented in Fig. 5, two different topologies can be considered.

In the strategy of reference (R1), each nodal series is individually predicted (which allows cross-nodal learning and cold-start forecasting). In the second one (R2), all nodes are jointly predicted for each time step of the horizon. These two topologies are tested on the six techniques described thereafter, resulting in 2*5 = 10 tested cases.

- a traditional *LSTM* recurrent neural network, composed of a single hidden layer.
- *R-DFFNN*, an hybrid forecaster merging a recurrent model with a deep feedforward neural network (DFFNN). The goal is to combine the strengths of a LSTM model (which is tailored to capture the sequential structure of DLMP sequences) with a regular fully connected feed-forward layer (which has the ability to learn relations among non-sequential data).
- *BLSTM*, a bidirectional LSTM which is dedicated to share information across all time steps of the horizon.
- *DBLSTM*, a deep architecture of BLSTM.
- *NS_DBLSTM*, a non spatial DBLSTM where location features and topology of the distribution system are not encoded so that there is no nodal price differentiation.

These forecasters are also compared with an ARIMA-GARCH model, where a standard Auto-Regressive Integrated Moving Average (ARIMA) model is combined with Generalized AutoRegressive Conditional Heteroscedasticity (GARCH) residuals, to leverage their ability to represent changes in variance over time. The confidence intervals around point forecasts is obtained from the variance derived with the GARCH model, by assuming a Gaussian distribution of the error [40]. Moreover, we also implement a naive methodology, consisting in partitioning the historical observations (in the training set) into 7*$|\mathcal{T}|$*$|\mathcal{N}|$ groups, respectively based on the day of the week, the time period of the day, and the node of the system. The empirical distribution within each group is then constructed, and is used (in the test set) as a naive benchmark to represent the uncertain DLMPs.

C. Hyper-parameters optimization

To make a fair comparison, we have carried out an extensive random search to find the combination of hyper-parameters that optimizes the predictive power of each model. Indeed, the forecaster has to be sufficiently sophisticated for reflecting the dynamics of nodal electricity prices, but not too complex for avoiding to overfit the model on the training observations, thus undermining its generalization capacity on unseen data.

The complexity of neural networks is defined by the number of hidden layers and the number of neurons within each layer. In deep models, different number of hidden layers are tested (between 2 and 6, since the first manual simulations have quickly shown that a higher number does not enhance the performance of the tools). The weights of neural networks were initialized using a Glorot uniform distribution. Also, the activation function of neurons from feedforward neural networks are rectified linear unit (ReLU).

In complement, early stopping is implemented for avoiding overfitting in the learning procedure. This consists in dividing the historical set of data into three sets, respectively for training, validating and testing. This allows stopping the learning phase (carried out on the training set) before the network begins to memorize the data instead of learning the underlying trend, on the basis of the model performance on unseen data (i.e. the validation set). At the end of the learning phase, the accuracy of the final model is evaluated on the test set.

D. Evaluation metrics

For the sake of completeness in performance comparison, all forecasters are also trained in a traditional deterministic framework. We evaluate the statistical quality of these point forecasts using the root mean square error (RMSE). This error metric focuses on the degree of correspondence between the deterministic predictions and the actual observations.

$$\text{RMSE} = \sqrt{\frac{\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}\left(\hat{y}_{t,n} - y_{t,n}\right)^2}{|\mathcal{N}| \times |\mathcal{T}|}} \tag{14}$$

where $\hat{y}_{t,n}$ is the deterministic output of the prediction model for client $n \in \mathcal{N}$ at time $t \in \mathcal{T}$, and $y_{t,n}$ is the actual value.

Then, four different probabilistic metrics are computed to evaluate the accuracy of the predicted quantiles. The performance is evaluated not only in terms of reliability (how closely the predicted intervals correspond to the actual data frequencies) and sharpness (concentration of the predicted intervals), but also through two global metrics, i.e., the pinball loss and the Winkler score, which quantify the compromise between these two criteria. Overall, sharpness and reliability need to be jointly analyzed, as high sharpness (i.e., a desirable property) is not always associated with a better prediction if the reliability of the model is low.

Firstly, we use a simple empirical measure of the reliability of the prediction intervals, by computing $\mathbb{E}\left(\mathbf{I}(y_{t,n} \leq \hat{y}^q_{t,n})\right)$ over the test set. The deviation with respect to the corresponding nominal probability q is a direct measure of forecast calibration.

Secondly, we evaluate the sharpness of the models using the prediction interval average width (PIAW), which is computed for a confidence interval of $(1-\alpha)\cdot 100\%$ as follows [41]:

$$\text{PIAW}_\alpha = \frac{1}{|\mathcal{T}|}\frac{1}{|\mathcal{N}|}\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}|\hat{y}^{1-\alpha/2}_{t,n} - \hat{y}^{\alpha/2}_{t,n}| \tag{15}$$

where $\hat{y}^{\alpha/2}_{t,n}$ and $\hat{y}^{1-\alpha/2}_{t,n}$ represent the $\alpha/2$ and $(1-\alpha/2)$ predicted quantiles for node n at time t.

Thirdly, as a first global metric, we use the total pinball loss, i.e. the average value of all pinball losses (11) across all quantiles (in this paper, q = 5, 10, 25, 50, 75, 90 and 95 %), over all points ($t \in \mathcal{T}, n \in \mathcal{N}$) of the space-time domain for each day of the test set. The smaller is the quantile loss, the better is the forecasting performance.

However, by averaging all quantiles in the final score, the total pinball loss may hide low reliability levels for extreme quantiles [42]. For instance, a high inaccuracy in the 5% quantile forecasts may have a limited impact on the global score. Hence, we complement the pinball loss with the Winkler score, which jointly quantifies if the intervals properly encapsulate the actual realization of uncertain variables (calibration), while considering the tightness of these intervals (sharpness), within a design where a lower score indicates a better probabilistic forecast. For a confidence interval of $(1-\alpha)\cdot 100\%$, the Winkler score is defined as:

$$\frac{1}{|\mathcal{T}|}\frac{1}{|\mathcal{N}|}\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}\begin{cases}\delta_{t,n} & L_{t,n}\leq y_{t,n}\leq U_{t,n}\\ \delta_{t,n}+2(L_{t,n}-y_{t,n})/\alpha & y_{t,n}<L_{t,n}\\ \delta_{t,n}+2(y_{t,n}-U_{t,n})/\alpha & y_{t,n}>U_{t,n}\end{cases} \tag{16}$$

where $L_{t,n}=\hat{y}_{t,n}^{\alpha/2}$ and $U_{t,n}=\hat{y}_{t,n}^{1-\alpha/2}$ are respectively the lower and upper bounds of the prediction interval (defined by the confidence level α), and $\delta_{t,n}=U_{t,n}-L_{t,n}$ is the interval width. If an observation $y_{t,n}$ falls into the predicted interval $[L_{t,n},U_{t,n}]$, the Winkler score is a direct measure of sharpness. Otherwise, a penalty term is added, whose value depends on the severity of the forecast error, hence integrating a calibration measure.

IV. Case study

This part intends to evaluate the performance of the different probabilistic forecasting tools. The neural networks have been implemented using Python 3.6.0 and the Keras library (along with the TensorFlow backend), whereas the Scikit-Learn tool has been employed for ensemble models (*QRF* and *GradBoost*). It should be noted that the same input data (Section II-C) are used by all models of the case study.

The benchmark is implemented for the IEEE European Low Voltage Test Feeder, shown in Fig. 6, which has a tree-structured topology composed of 6 line ramifications. The original network, which is used to compare the different forecasters in Section IV-A, feeds a total of $|\mathcal{N}_1| = 57$ nodes, among which 55 are residential prosumers with single-phase connections, while 2 are community-scale PV plants (100 kWp) with three-phase connections. Among residential clients, 40 are equipped with PV sources (6 kWp) and batteries (4 kW, 8 kWh), while the other 15 have only inflexible loads. In order to evaluate the ability of cold-start forecasting, a 3-phase 100kW industrial load 'Ind. Load' (node D) is added in Section IV-B.

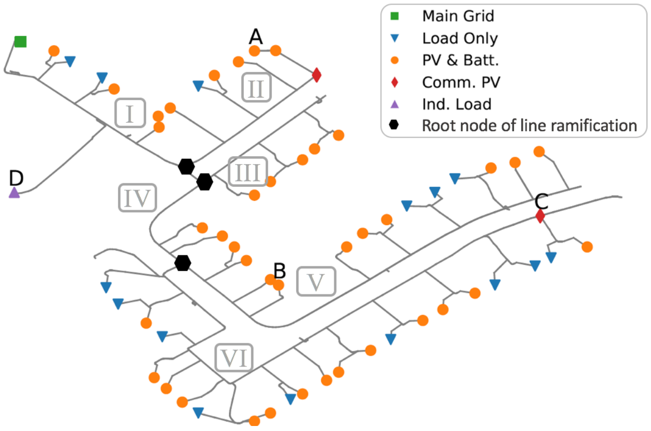

Fig. 6. The case study distribution network, composed of 6 line ramifications (I to VI) feeding 57 clients, i.e. 15 single-phase inflexible loads, 40 single-phase prosumers (with inflexible load, PV source and battery system) and 2 three-phase community-scale PV plants. The industrial load (node D) is added to the (resulting 58-bus) system at a later stage.

The DLMPs database comes from the three-phase probabilistic dispatch in [3], realized based on residential load and PV generation data from the Customer-Led Network Revolution Trial from October 2012 to March 2014 [43].

A. Comparison of models for DLMPs forecasting

The DLMPs are predicted over a (daily) multi-horizon of $|\mathcal{T}| = 48$ intervals of 30 minutes for the $|\mathcal{N}_1| = 57$ nodes. The available database has a total of 478 days, among which 286 and 96 are respectively used for model training and validation, while the last 96 are applied for model testing. Table I presents the performance of the different (naive, static and recurrent) tools, and their respective ability to capture the space-time dependencies in DLMPs. Practically, we compare the static

TABLE I
Overall performance of different methods for probabilistic DLMPs forecasting.

Topology	Model	RMSE	Pinball loss	Winkler score [c£/kWh]			Empirical coverage [%]			
		[c£/kWh]	[c£/kWh]	α = 0.1	α = 0.2	α = 0.5	q = 5%	q = 25%	q = 75%	q = 95%
naive methodology		2.52	3.37	9.97	8.18	5.12	3.4	21.1	73.7	94.1
ARIMA-GARCH		2.02	2.47	7.23	5.84	3.97	6.2	27.0	75.0	94.2
static (S1)	*FFNN*	1.49	2.14	5.57	4.70	3.41	3.2	19.9	74.4	95.8
static (S1)	*DFFNN*	1.31	1.91	4.99	4.20	3.05	3.9	21.0	72.6	93.6
static (S1)	*QRF*	1.36	2.28	6.45	5.19	3.57	11.8	30.6	65.1	86.4
static (S1)	*GradBoost*	1.50	2.14	5.72	4.74	3.40	3.1	23.7	73.0	95.1
static (S2)	*DFFNN*	1.43	2.04	5.79	4.55	3.21	7.7	22.2	72.4	92.4
static (S3)	*DFFNN*	1.55	2.18	5.90	4.98	3.47	3.0	21.7	74.5	96.6
static (S4)	*DFFNN*	1.62	2.22	6.65	5.24	3.39	6.1	25.7	71.6	92.5
recurrent (R1)	*LSTM*	1.42	1.69	4.38	3.74	2.70	4.3	25.0	75.6	94.2
recurrent (R1)	*R-DFFNN*	1.27	1.62	4.29	3.59	2.56	4.6	24.0	75.5	96.1
recurrent (R1)	*BLSTM*	1.20	1.33	3.63	2.97	2.09	6.7	26.8	70.1	91.5
recurrent (R1)	*DBLSTM*	0.93	1.19	3.20	2.65	1.88	4.7	24.1	68.0	91.4
recurrent (R1)	*NS_DBLSTM*	1.25	1.40	3.90	3.24	2.27	5.3	26.3	71.4	91.5
recurrent (R2)	*DBLSTM*	1.22	1.31	4.00	2.99	2.01	4.5	22.9	72.5	93.7

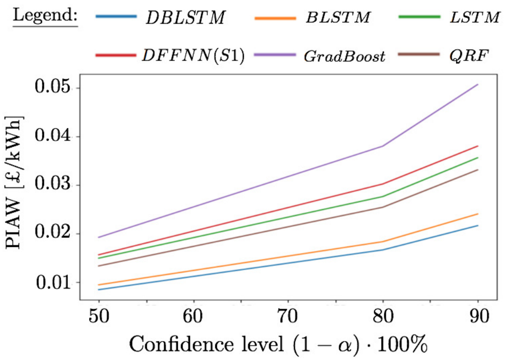

Fig. 7. Sharpness estimate using the prediction interval average width (PIAW).

forecasting techniques presented in Section III-A on the single-output topology (S1). Then, we select the best technique (i.e. the deep feedforward neural network), and we apply it on the other three (multi-output) topologies (S2, S3 and S4). Similarly, we compare the different recurrent models on the single output topology (R1), which highlights the advantages of the bidirectional processing of data in multi-step forecasting as well as the effect of spatial information on the prediction performance. Finally, we compare this learning framework (where each nodal price series is treated individually) with the multi-output topology (R2). This information is completed in Fig. 7, where the average sharpness for nominal coverage rates $(1-\alpha)\cdot 100\% \in [50-90]\%$ is depicted for different models.

The experiments show that all neural networks strongly outperform both the naïve (averaging) method and the ARIMA-GARCH statistical model. This observation is aligned with previous studies [22], [44], which show that the linear nature of econometric models make them poorly suited for predicting the highly nonlinear behavior (and quick fluctuations) of the price signals. This trend is exacerbated by the increasing penetration of renewable sources, since prices are becoming more volatile with frequent price spikes that require advanced nonlinear tools to be accurately forecasted.

Interestingly, we also observe that LSTM-based recurrent neural networks are well-calibrated, i.e., $\mathbb{P}\left(y_{t,n} \leq \hat{y}_{t,n}^{q} \mid x_{t,n}\right) \approx q$, and exhibit higher sharpness than other models, with the exception of the *QRF*. Indeed, the quantile random forest yields narrow quantiles (leading to high sharpness as depicted in Fig. 7), but this does not contribute to improved forecast accuracy in comparison to other models such as deep feedforward neural networks and traditional LSTM that consistently achieve higher performances. Generally, static tools (*QRF*, *GradBoost* and feedforward neural networks) are structure agnostic, which makes them broadly applicable, but at the expense of a weaker performance than recurrent models which are purposefully tailored to the time structure of the multi-step prediction [45].

This observation is strengthen by the improvement associated with the bidirectional LSTM architecture (relative increase in accuracy of around 20% with respect to the standard LSTM). This difference illustrates the BLSTM ability to make more use of surrounding context than the other tools, by efficiently sharing the information among the different time periods of the prediction horizon.

Acting in a complementary way, the advantage of deep networks is also obvious. In line with the current literature, we see that deep architectures (with several hidden layers) have better generalization capabilities than shallow ones, with the pinball loss dropping from 1.33 to 1.19 c£/kWh as the number of layers increases from one to five for the BLSTM model. In the latter model, which is the most efficient of the benchmark, each hidden layer is composed of 20 neurons. A similar improvement of approximately 10%, from 2.14 to 1.91 c£/kWh, is observed for feedforward networks (in which the optimal complexity is obtained with four hidden layers). However, in accordance with [22], the hybrid R-DFFNN model exhibits comparable performances than its individual components, which illustrates the difficulty to design hybrid architectures that improve the global prediction accuracy. Overall, the best performance, in terms of both reliability and sharpness, is achieved by the deep bidirectional LSTM model. Its hyper-parameters are presented in Table II, which is complemented in Table III with a sensitivity analysis evaluating the effects of the model complexity on the related accuracy. Different numbers of neurons within hidden layers are tested, i.e., {10, 20, 50, 100, and 200}, but the same number of neurons is used for forward and backward layers in the proposed DBLSTM tool.

TABLE II
HYPER-PARAMETERS OF THE DBLSTM MODEL.

Hyper-parameter	Value
number of layers (L)	5
number of neurons by layer	20
Training algorithm	Adam
Batch size	16
Time lag	0
Regularization	Early stopping

The optimal structure is a DBLSTM layer with 20 neurons in each of the 5 hidden layers. The batch size to train the model is set to 16 daily sequences (of 48 half-hourly intervals) for the Adam algorithm. This stochastic gradient descent procedure endogenously adapts the learning rate during the learning. Due to the nature of the task, in which predictions are needed with a horizon of interest ranging from 12 to 36 hours from the forecast creation time (Fig. 1), no past data are treated by the model. Also, early stopping proves sufficient to achieve good generalization capabilities, and traditional regularization techniques such as dropout or adding penalty terms using L1-L2 norms in the loss function (to enforce sparsity on network weights) do not lead to improved results. Finally, to deal with differences in the scales across (input and output) variables, all data are individually standardized, using a robust scaler that removes the median and scales the data according to the quantile range [0.1, 0.9].

From Table III, we see that increasing the model complexity is beneficial but may ultimately lead to overfitting issues (due to network parameter redundancy). Also, we observe that relying on additional hidden layers is more efficient in improving performance than adding more neurons within the same recurrent layer. However, it should be reminded that the hyper-parameter solution is closely linked to the size of the training database. In the case of a limited dataset, a more

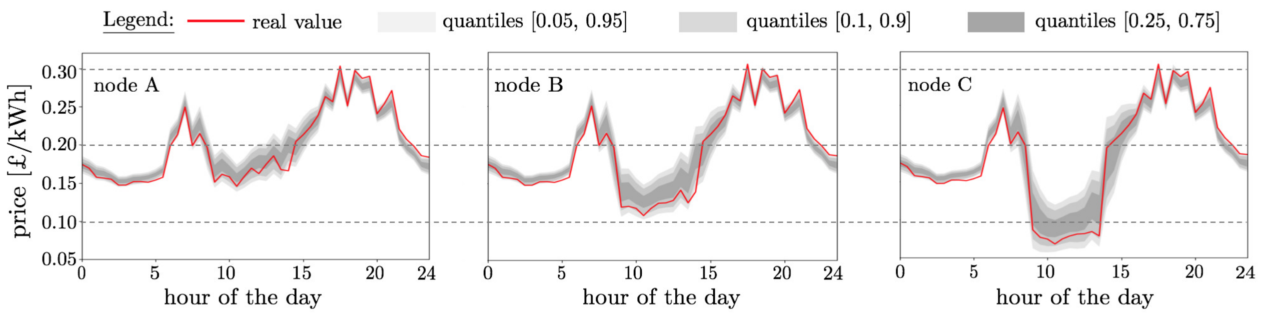

Fig. 8. Day-ahead probabilistic forecasts of the DLMPs associated with three different nodes (A, B and C in Fig. 6) of the distribution system.

TABLE III
SENSITIVITY ANALYSIS ON THE COMPLEXITY OF THE DBLSTM MODEL.

Model complexity # layers - # neurons	Pinball loss [c£/kWh]
1 - 100	1.33
2 - 100	1.27
3 - 50	1.23
4 - 20	1.20
5 - 20	1.19
6 - 20	1.23

compact (less complex) model is more likely to be selected to avoid a model overfitting on the limited information.

Fig. 8 shows the day-ahead DLMP forecasts of 3 nodes (A, B, and C in Fig. 6) during a summer day (subject to a base load with high PV generation) for the best model (DBLSTM). The gray areas stand for the forecasted quantiles and the red line denotes the actual price series. At this stage, it is important to remind that DLMPs are defined by three main components: energy, losses, and voltage violations. In general, uncertainties on the energy component arise from the underlying load and generation uncertainties. Here, we notice that such forecast uncertainties are globally small in comparison with the average price of around 20 c£/kWh, and that the quantiles can effectively seize the variability in the DLMP profiles. However, during periods of peak PV generation and low consumption (in the middle of the day), voltage violations (and increased losses) are observed, resulting in lower price values. Since the actual voltage violation rates are highly uncertain (and thus difficult to predict), this leads to increased uncertainty on nodal prices during these off-peak periods (which is associated with larger confidence intervals on DLMP values). Also, these violations of the network voltage limits on certain buses lead to nodal price differentiation (mainly between 08:00 am and 16:00 pm). Interestingly, we see that the forecaster has properly captured these spatial dependencies, thereby suggesting that the proposed approach is effective in learning across the related nodal price series.

To further illustrate the contribution of the energy and voltage components in the forecasted DLMPs, we show in Fig. 9 the predictions for node A for a classical autumn day (during which the probability of stressed network conditions is close to 0). During this day, the DLMP uncertainty is mainly driven by the energy component, and we observe that price uncertainty is higher during peak periods (slightly in the morning and more prominently in the evening).

Over the test set, no congestion nor voltage violation (and thus no price discrepancies) were observed during night periods, which explains the acceptable global accuracy of the non-spatial *NS_DBLSTM* model in which all nodes have roughly the same price profile. Based on these outcomes, we see that the predicted DLMPs provide useful day-ahead information incentivizing the prosumers to increase their individual self-consumption when the price of energy is low during the middle of the day (to alleviate overvoltage issues), and to flatten their load when the price of energy is highest, hence leading to safer operating conditions.

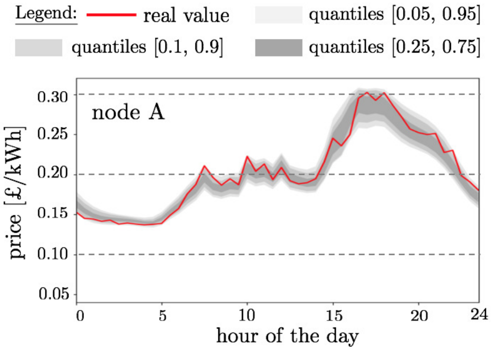

Fig. 9. Day-ahead probabilistic forecasts of daily DLMPs of node A for a working day in autumn.

At this stage, it is important to notify that, for achieving decent accuracy on the multi-output architectures (S2, S3, S4 and R2), a different model had to be trained for each $q \in \mathcal{Q}$. This operation is needed to reduce the size of the output space of each individual model, which is realized at the expense of an increased computational burden. In general, these multi-output models attempt to better account for correlations between outputs, but are inevitably plagued with scalability issues, e.g. model (S4) requires to compute $|\mathcal{T}|*|\mathcal{N}_1|$ = 48*57 = 2736 points (for each q) in a single instance, which is impractical in view of the number of historical data. This leads to an increase of the pinball loss from 1.91 (for model S1) to 2.22 c£/kWh (i.e. accuracy loss of 16%). Likewise, the joint prediction of all nodal price series with the DBLSTM (R2) is associated with a drop in performance of 21% in comparison with the single-output variant (R1). Overall, the outcomes show that relying on a compact model (with a limited dimensionality of both input and output feature spaces) is a key element to extract the predictive power of machine learning techniques.

B. Cold-start DLMP forecasting

By considering each nodal DLMP individually (in R1), the DBLSTM is able to smoothly accommodate nodes with little history (without modifying the tool structure nor relying on uncertain inference strategies). This appealing feature is further investigated by adding a large 3-phase 100 kW industrial load (node D) in the case study. We then perform the day-ahead probabilistic forecast for all $|\mathcal{N}_2| = 58$ nodes over the same 96 days of the test set.

The prediction outcome for the new node D is depicted in Fig. 10 (for the first week of the test set). It is observed that the proposed tool has powerful generalization capacities which enables to efficiently determine the price profile of the new node. However, the size of the industrial load has a significant impact on the power flows within the system, and thus on the resulting DLMPs. The trained tools are therefore not well-calibrated to these new conditions, with the global pinball loss of the DBLSTM growing from 1.19 to 2.62 c£/kWh, mainly due to a loss of sharpness coming from the increased uncertainty. Interestingly, the *NS_DBLSTM* suffers a greater deterioration with the pinball loss increasing to 2.88 c£/kWh, which shows that our generic way used to encode spatial information is not only efficient in capturing nodal discrepancies, but makes also the tool more resilient to changes in the operating conditions. Overall, the ill-conditioning effects can be smoothly alleviated over time through a proper recalibration of the model (that can be progressively fitted to the updated system topology) with the new data revealed each day [46]-[48].

V. Conclusion

This paper addresses the problem of a distribution system operator who is responsible to come up with probabilistic forecasts of DLMPs in low-voltage systems for incentivizing end-users to account for network constraints in their prosumption profile. These DLMPs are not only correlated across consecutive time steps, but also among the different nodes of the distribution network (due to technical limitations). We show that the combination of deep, bidirectional Long Short-term Memory RNNs (to capture time information) with a strategy dedicated to learn across nodal price series gives state-of-the-art results in probabilistic DLMP prediction.

The main advantage of the method is to capture space-time dependencies in a framework inherently scalable to large systems thanks to an unified treatment across all nodes of the distribution network, which moreover ensures adaptability to fluctuating operating conditions. Moreover, the strategy allows to smoothly accommodate new end-users without historical information (i.e. cold-start forecasting) by leveraging the relevant information from other nodes. Additionally, the generic nature of the methodology can be applied to other space-time problems such as forecasting transmission LMPs, or the localized probability of voltage violation or line congestion within power systems.

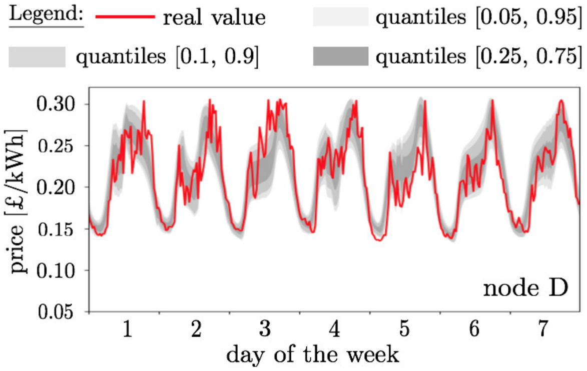

Fig. 10. Cold-start prediction of the new client connected to the system.

References

[1] Y. Parag and B. K. Sovacool, "Electricity Market Design for the Prosumer Era," *Nature Energy*, vol. 1, no. 4, p. 16032, 2016.

[2] A. Papavasiliou, "Analysis of Distribution Locational Marginal Prices," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4872–4882, Sept. 2018.

[3] T. Morstyn, A. Teytelboym, C.Hepburn and M. D. McCulloch, "Integrating P2P Energy Trading with Probabilistic Distribution Locational Marginal Pricing," *IEEE Trans. Smart Grid*, 2020.

[4] C. Edmunds, S. Galloway and S. Gill, "Distributed electricity markets and distribution locational marginal prices: A review," 52nd International Universities Power Engineering Conference (UPEC), Heraklion, pp. 1-6, 2017.

[5] R. Tabors, M. Caramanis, E. Ntakou, G. Parker, M. Van Alstyne, P. Centolella, and R. Hornby, "Distributed energy resources: New markets and new products," Proceedings of the 50th Hawaii International Conference on System Sciences, 2017.

[6] K. Zheng, Y. Wang, K. Liu and Q. Chen, "Locational Marginal Price Forecasting: A Componential and Ensemble Approach," *IEEE Trans. Smart Grid*, in press.

[7] A. Radovanovic, T. Nesti and B. Chen, "A Holistic Approach to Forecasting Wholesale Energy Market Prices," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4317-4328, Nov. 2019.

[8] L. Bai, et. al, "Distribution Locational Marginal Pricing (DLMP) for Congestion Management and Voltage Support," *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 4061-4073, July 2018.

[9] Y. Ji, R. J. Thomas and L. Tong, "Probabilistic Forecasting of Real-Time LMP and Network Congestion," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 831-841, March 2017.

[10] J. Lago, F. De Ridder, P. Vrancx, B. De Schutter, "Forecasting day-ahead electricity prices in Europe: The importance of considering market integration," *Applied Energy*, vol. 211, pp. 890–903, 2018.

[11] M. Sun, C. Feng, J. Zhang, "Conditional aggregated probabilistic wind power forecasting based on spatio-temporal correlation," *Applied Energy*, vol. 256, Dec. 2019.

[12] P. Pinson and H. Madsen, "Adaptive modelling and forecasting of offshore wind power fluctuations with Markov-switching autoregressive models," *J. Forecast.*, vol. 31, no. 4, pp. 281–313, 2012.

[13] Y. Goude, R. Nedellec and N. Kong, "Local Short and Middle Term Electricity Load Forecasting With Semi-Parametric Additive Models," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 440-446, Jan. 2014.

[14] G. H. Bai, B. Fleck, and M. J. Zuo, "A stochastic power curve for wind turbines with reduced variability using conditional copula," *Wind Energy*, vol. 19, no. 8, pp. 1519–1534, Aug. 2015.

[15] J. Tastu, P. Pinson, P.-J. Trombe, and H. Madsen, "Probabilistic forecasts of wind power generation accounting for geographically dispersed information," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 480–489, Jan. 2014.

[16] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, "DeepSaliency: multi-task deep neural network model for salient object detection," *IEEE Trans. Image Process*, vol. 25, no. 8, pp.3919-3930, 2016.

[17] A. Ghaderi, B. M. Sanandaji, and F. Ghaderi, "Deep Forecast: Deep Learning-based Spatio-Temporal Forecasting." arXiv preprint arXiv: 1707.08110, 2017.

[18] Z. Yang, L. Ce, L. Lian, "Electricity price forecasting by a hybrid model, combining wavelet transform, ARMA and kernel-based extreme learning machine methods," *Applied Energy*, vol. 190, pp. 291-305, 2017.

[19] J. Dowell and P. Pinson, "Very-Short-Term Probabilistic Wind Power Forecasts by Sparse Vector Autoregression," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 763-770, March 2016.

[20] Y. Zhao, L. Ye, P. Pinson, Y. Tang and P. Lu, "Correlation-Constrained and Sparsity-Controlled Vector Autoregressive Model for Spatio-Temporal Wind Power Forecasting," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5029-5040, Sept. 2018.

[21] F. Golestaneh, P. Pinson, R. Azizipanah-Abarghooee and H. B. Gooi, "Ellipsoidal Prediction Regions for Multivariate Uncertainty Characterization," *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 4519-4530, July 2018.

[22] J. Lago, F. De Ridder, and B. De Schutter, “Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms,” *Applied Energy*, vol. 221, pp. 386–405, 2018.

[23] M. Sun, T. Zhang, Y. Wang, G. Strbac and C. Kang, ”Using Bayesian Deep Learning to Capture Uncertainty for Residential Net Load Forecasting,” *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 188-201, Jan. 2020.

[24] Y. Wang, N. Zhang, Q. Chen, D. S. Kirschen, P. Li, and Q. Xia, “Data-driven probabilistic net load forecasting with high penetration of behind-the-meter PV,” *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3255–3264, May 2018.

[25] 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, March 2019.

[26] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. Hyndman, “Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond,” *International Journal of Forecasting*, vol. 32, no. 3, pp. 896–913, 2016.

[27] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C Woo, ”Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” *Advances in neural information processing systems*, pp. 802–810, 2015.

[29] M. Khodayar and J. Wang, ”Spatio-Temporal Graph Deep Neural Network for Short-Term Wind Speed Forecasting,” *IEEE Trans. Sustain. Energy*, vol. 10, no. 2, pp. 670-681, April 2019.

[30] R. Wen, K. Torkkola, and B. Narayanaswamy, “A multi-horizon quantile recurrent forecaster,” unpublished paper, 2017. [Online]. Available: https://arxiv.org/abs/1711.11053v1.

[31] B. Uniejewski, R. Weron and F. Ziel, ”Variance Stabilizing Transformations for Electricity Spot Price Forecasting,” *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 2219-2229, March 2018.

[32] J.-F. Toubeau, J. Bottieau, F. Vallée, and Z. De Grève, “Improved day-ahead predictions of load and renewable generation by optimally exploiting multi-scale dependencies,” *Proc. 7th IEEE Conf. Innovative SmartGrid Technol.*, Dec. 2017.

[33] A. Graves and J. Schmidhuber, ”Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” *Neural Networks*, vol. 18, no. 5–6, pp. 602–610, July 2005.

[34] A. Graves, A. Mohamed, G. Hinton, ”Speech Recognition with Deep Recurrent Neural Networks,” ICASSP 2013, Vancouver, Canada, 2013.

[35] Y. Wang, Q. Hu, D. Meng, and P. Zhu, “Deterministic and probabilistic wind power forecasting using a variational Bayesian-based adaptive robust multi-kernel regression model,” Appl. Energy, vol. 208, pp. 1097–1112, 2017.

[36] Y. Liu, H. Qin, Z. Zhang, et al.: ‘Ensemble spatiotemporal forecasting of solar irradiation using variational Bayesian convolutional gate recurrent unit network’, Appl. Energy, vol. 253, p. 113596, Nov. 2019.

[37] Y. Liu, H. Qin, Z. Zhang, S. Pei, Z. Jiang, and Z. Feng, “Probabilistic spatiotemporal wind speed forecasting based on a variational bayesian deep learning model,” Appl. Energy, vol. 260, 2020, doi: 10.1016/j.apenergy.2019.114259.

[38] R. Koenker and B. Jr Gilbert, “Regression quantiles,” Econometrica, J. Econometric Soc., vol. 46, pp. 33–50, 1978.

[39] Y. Wang, D. Gan, M. Sun, N. Zhang, Z. Lu, and C. Kang, “Probabilistic individual load forecasting using pinball loss guided lstm,” *Applied Energy*, vol. 235, pp. 10–20, Feb. 2019.

[40] M. David, F. Ramahatana, P.-J., Trombe, et al. ‘Probabilistic forecasting of the solar irradiance with recursive ARMA and GARCH models’, Sol. Energy, vol. 133, pp. 55–72, 2016.

[41] M. Khodayar, S. Mohammadi, M. E. Khodayar, J. Wang and G. Liu, ”Convolutional Graph Autoencoder: A Generative Deep Neural Network for Probabilistic Spatio-Temporal Solar Irradiance Forecasting,” *IEEE Trans. Sustain. Energy*, vol. 11, no. 2, pp. 571-583, April 2020.

[42] J. Bottieau, L. Hubert, Z. De Grève, F. Vallée, 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, March 2020.

[43] Customer-Led Network Revolution, “Enhanced Profiling of Domestic Customers With Solar Photovoltaics.” [Online]. Available: networkrevolution.co.uk.

[44] N. Amjady and M. Hemmati, “Energy price forecasting-problems and proposals for such predictions,” IEEE Power Energy Mag., vol. 4, no. 2, pp. 20–29, Mar./Apr. 2006.

[45] C. Feng, M. Sun and J. Zhang, ”Reinforced Deterministic and Probabilistic Load Forecasting via Q-Learning Dynamic Model Selection,” *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1377-1386, March 2020.

[46] K. Hubicka, G. Marcjasz and R. Weron, ”A Note on Averaging Day-Ahead Electricity Price Forecasts Across Calibration Windows,” *IEEE Trans. Sustain. Energy*, vol. 10, no. 1, pp. 321-323, Jan. 2019.

[47] J.-F. Toubeau, P.-D. Dapoz, J. Bottieau, A. Wautier, Z. De Grève, F. Vallée, ”Recalibration of Recurrent Neural Networks for Short-Term Wind Power Forecasting,” *Electric Power Systems Research*, vol. 190:106639, pp. 1-7, Jan. 2021.

[48] A. Bracale, P. Caramia, P. De Falco and T. Hong, ”Multivariate Quantile Regression for Short-Term Probabilistic Load Forecasting,” *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 628-638, Jan. 2020.

Jean-François Toubeau (M’18) 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 post-doctoral researcher of the Belgian Fund for Research (F.R.S/FNRS) within the ”Power Systems and Markets Research Group” of the same University. He was a visiting researcher at KU Leuven from September 2019 to February 2020. His research mainly focuses on bridging the gap between machine learning and decision-making in modern power systems.

Thomas Morstyn (M’16) received the B.Eng. degree (Hons.) in electrical engineering from the University of Melbourne in 2011, and the Ph.D. degree in electrical engineering from the University of New South Wales in 2016. He is a Lecturer in power electronics and smart grids with the School of Engineering, University of Edinburgh. He is also a Visiting Fellow with the Oxford Martin School, University of Oxford. His research interests include multiagent control and market design for integrating distributed energy resources into power system operations.

Jérémie Bottieau (S’17) received the Diploma in electrical engineering from he University of Mons, Belgium, where he has been working toward the Ph.D. degree with the ”Power Systems and Markets Research Group” since 2017. His research interests include short-term forecasting and optimization in electricity markets.

Kedi Zheng (S’17) received the B.S. degree in electrical engineering and automation from the Department of Electrical Engineering, Tsinghua University, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include application of big data analytics for electricity market.

Dimitra Apostolopoulou (M’11) was awarded a Ph.D. and a M.S. in Electrical and Computer Engineering from University of Illinois at Urbana-Champaign in 2014 and 2011, respectively. She received her undergraduate degree in Electrical and Computer Engineering from National Technical University of Athens, Greece in 2009. She is currently a Lecturer at City, University of London. Previously, she was a Postdoctoral researcher at University of Oxford and a Lecturer at Christ Church College. Priorly, she worked at the Smart Grid and Technology Department at Commonwealth Edison Company. Her research interests include power system operations and control, market design and economics.

Zacharie De Grève (M'12) received Electrical and Electronics Engineering degree in 2007 from the Faculty of Engineering of Mons, University of Mons, Belgium, where he received the Ph.D. degree in electrical engineering in 2012. He was a Research Fellow with the Belgian Fund for Research (F.R.S/FNRS) until 2012. He is currently a Researcher with the "Power Systems and Markets Research Group" at the University of Mons, and a part-time Lecturer since September 2019. He conducts transverse research in machine learning, optimization and energy economics, applied to modern electricity networks with a high share of renewables, in order to contribute to the energy transition.

Yi Wang (M'19) received the B.S. degree from the Department of Electrical Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2014, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2019. He was also a Visiting Student Researcher with the University of Washington, Seattle, WA, USA from 2017 to 2018. He is currently a Postdoctoral Researcher with ETH Zürich. His research interests include data analytics in smart grid and multiple energy systems.

François Vallée (M'09) received the degree in civil electrical engineering and the Ph.D. degree in electrical engineering from the Faculty of Engineering, University of Mons, Belgium, in 2003 and 2009, respectively. He is currently an Associate Professor and leader of the "Power Systems and Markets Research Group" at the University of Mons. His Ph.D. work has been awarded by the SRBE/KBVE Robert Sinave Award in 2010. His research interests include PV and wind generation modeling for electrical system reliability studies in presence of dispersed generation. He is currently a member of the Governing Board from the 'Société Royale Belge des Electriciens - SRBE/KBVE' (2017) and an Associate Editor of the International Transactions on Electrical Energy Systems (Wiley).