Probabilistic day-ahead energy price forecast by a Mixture Density Recurrent Neural Network

Alessandro Brusaferri[a,b], Matteo Matteucci[b], Danial Ramin[a], Stefano Spinelli[a,b], Andrea Vitali[a]
[a]*CNR-Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, Milan, Italy*
[b]*Politecnico di Milano, Milan, Italy*
$name.surname$@stiima.cnr.it, $name.surname$@polimi.it

***Abstract*—Probabilistic electricity price forecast (EPF) systems represent a fundamental tool to achieve robust production scheduling and day-ahead bidding strategies. However, most EPF methods, including recently proposed deep learning based techniques, are still targeting point predictions, following the common Gaussian assumption. In this work, we propose a novel probabilistic EPF approach based on the integration of a Gaussian Mixture layer, parametrized by a Recurrent Neural Network with Gated Recurrent Units, including an L1-norm based feature selection mechanisms. The network is conceived to approximate general conditional price distributions through learning. Moreover, we developed a multi-hours prediction approach exploiting correlations and patters both in hourly and cross-hour contexts. Experiments have been performed on the Italian market dataset, showing the capability of the proposed method to achieve accurate out-of-sample predictions while providing explicit uncertainty indications supporting enhanced decision making.**

***Index Terms*—Electricity markets, Price forecast, Probabilistic Forecast, Recurrent Neural Network, Gaussian Mixture Model**

I. Context and motivation

The last two decades have witnessed a dramatic change in energy markets and policies. Market deregulation has been pushed to improve efficiency of the electricity supply. Contextually, balancing power markets have been expanded to increase the security of the transmission system. Among these emerging competitive markets are the day-ahead markets where the price is cleared hourly, based on the received generation and demand bids conditioned on the equilibrium between the two. Being able to predict the hourly price allows the participant of these markets to exploit new opportunities and increase their profit. However, accurate price prediction is very challenging, and the greater is the uncertainty on the predicted price, the lager is the risk margin [1]. The main issue of electric energy regulation is that, unlike other commodities, electricity is not storable. Thus, production-consumption equilibrium has great implications for grid security. Several exogenous factors make the prediction hourly prices challenging [2]. Demand patterns are changing following the increasing market involvement. Production is becoming more volatile due to the increasing penetration of renewable energies. All these factors have led the electricity price to be strongly nonlinear with non-stationary mean and variance, in addition to high seasonality. Consequently, a substantial scientific effort has been devoted to developing forecasting tools; a detailed review is reported in [3]. While fundamental and multi-agent models are often employed for strategic analysis and long-term predictions of markets dynamics, statistical and computational intelligence-based techniques have been proved to be most effective for prediction of short term electricity price [4]. While both are capable of capturing the nonlinear behavior of the energy price, computational intelligence techniques have been increasingly proved effective in handling complex dynamics such as the one presented in day-ahead markets. Among those, deep neural network architectures have gained considerable attention due to their capability in extracting hierarchical features from the data [5]. A comprehensive study on the state-of-the-art implementation of these models can be found in [6].

Regardless of the model applied, almost all the previous studies have been focused on point forecasts [1]. However, a probabilistic model capable of providing insight into the uncertainty in the prediction can be an invaluable tool both for generation companies and large-scale consumers who deal with the various source of stochasticity in their processes and need to account for risk in the decision-making chain. To the best of our knowledge, the only previous work exploring modern deep neural networks in a probabilistic framework is [7], but still in a simplified Gaussian distribution assumption. Indeed, as stated in the recent review [1], probabilistic EPF is a "fascinating but still underdeveloped" field.

In this paper, we extend the aforementioned developments by proposing a neural network-based probabilistic EPF approach beyond the Gaussian distribution assumption. To this end, we introduce a Gaussian Mixture output layer to approximate generic conditional distributions. Then, we parametrize the mixture by a recurrent neural network with Gated Recurrent Units, aimed to extract useful features from multi-input sequences including the exogeneous variables, summarized within the latent state. Also, we develop a multi-period forecasts approach employing only input data available in real bidding conditions, including a L1-norm based automated feature selection mechanism, to identify correlations and patterns both in hourly and cross-hour contexts. Our method is conceived to intrinsically tackle heteroskedasticity, by learning input conditioned variances on each hour constituting the day-ahead prediction horizon.

We have compared our method to a deterministic RNN and a Bayesian Neural Network on the Italian day-ahead market datasets, showing increases in prediction accuracy as well as the capability to provide forecast distributions.

The paper is structured as follows. Section II starts introducing the EPF problem following the conventional Gaussian assumption, as typically adopted in point forecast methods. Then, the proposed approach is described, covering the approximation of broader classes of conditional distributions by a mixture layer, the architecture design and the developed training method. Section III reports the results achieved.

II. EPF Method

Neural EPF models providing point forecast are typically learned by minimizing the sum of squares error over the training data set. By adopting such EPF framework, the conditional distribution of the energy prices given the input variables set is often implicitly assumed to be Gaussian [1], thus leading to the following expression:

$$P_\theta(y_{t+1}|x_t) = \frac{1}{\sqrt{2\pi\sigma_y}} e^{-\frac{1}{2\sigma_y{}^2}(\hat{f}_\theta(x_t)-y_{t+1}))^2} \quad (1)$$

where $y_{t+1} \in \mathbb{R}^{n_y}$ represents the target prices whereas $\hat{f}_\theta$ is the employed forecasting network, parametrized by $\theta \in \mathbb{R}^{n_\theta}$. $x_t \in \mathbb{R}^{n_x}$ comprises the current and past price values, as well as current and past values of a chosen set of input conditioning variables, such as load demand, solar/wind predictions, etc.

From a statistical perspective, the main goal of EPF network training is to achieve the best approximate representation of the unknown underlying generator, thus supporting forecast in test conditions. Following this Gaussian assumption, statistics of the target data are achieved by learning the best - often sub-optimal - approximation of the conditional mean through the network, followed by the estimation of the variance parameter. This is obtained in practice by minimizing the negative log-likelihood of the available observations. Therefore, commonly adopted neural EPF models approximate the conditional average of the prices in the dataset, as a function of the input data, through the network parametrization at the local minimum of the loss function reached during training. Afterwards, a global variance parameter is calculated from the residuals using the network predictions (i.e., as average prediction variance) as follows:

$$\hat{\sigma}_y^2 = \frac{1}{S}\sum_{s=1}^{S}\left[\hat{f}_\theta\left(\mathbf{x}^{(s)};\theta^*\right) - y^{(s)}\right]^2 \quad (2)$$

where we calculate the variance $\hat{\sigma}_y^2$ over the whole length of the data set. S is the number of samples in the training set and θ^* the learned network parameters values. It is worth noting that the exploitation of neural EPF models trained by sum of squares loss does not strictly require the underlying distribution to be Gaussian. Nevertheless, the network cannot distinguish it from other distributions characterized by the same statistics [8].

In practical situations, the price distribution to be identified strictly depends on the specific characteristics of the energy market under treatment, thus requiring probabilistic models overcoming the Gaussian assumption. Hence, we developed a neural EPF architecture aimed to model a broader class of conditional distributions, as detailed in the following section.

A. Probabilistic EPF network architecture

The developed EPF network architecture is represented in Figure 1. First of all, we replaced the linear layer employed in previous neural EPF models with a Mixture layer, following the Mixture Density Network (MDN) concept. Representing more a class of techniques than a specific network design, MDN has been introduced in the seminal work of [8] by stacking a feedforward neural network (FFNN) with a Mixture layer, to develop machines capable to approximate conditional distributions.

In MDNs, the neural network is employed to learn the parameters of the Mixture model. The approach has been demonstrated in [8] on a toy problem to map the inverse kinematics of a simple 2-link robot arm. Afterwards, MDNs have been investigated in several application fields including text generation [9], trajectory predictions [10] and games [11], showing promising results. Nevertheless, to the best of our knowledge, MDNs are still not explored within the day-ahed energy price forecast research field.

Starting from the general MDN concept, several neural EPF models can be conceived. A first design choice regards the specification of the kernels and covariance matrices structure. Several alternatives have been proposed in the literature (see e.g. [9], [12], [13]). In this work, we employed a spherical Gaussian kernel, characterized by a common variance parameter within each mixture component. In this way, we avoid the computational expense of full covariance matrices while still supporting the capability to approximate the underlying den-

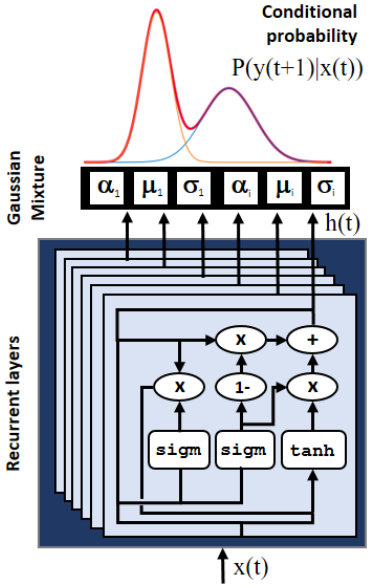

Fig. 1. Probabilistic EPF network

sity function to arbitrary accuracy [14]. ALso, the statistical independence assumption of outputs is avoided, as opposed to the Gaussian formalization performed in (1). Formally, the spherical Gaussian kernel is defined as:

$$\phi_k(\mathrm{y}|\mathrm{x}) = \frac{1}{(2\pi)^{n_y/2}\sigma_k(\mathrm{x})^{n_y}} \exp\left\{-\frac{\|\mathrm{y}-\mu_k(\mathrm{x})\|^2}{2\sigma_k(\mathrm{x})^2}\right\} \quad (3)$$

where $\mu_k(\mathrm{x}) \in \mathbb{R}$ and $\sigma_k(\mathrm{x}) \in \mathbb{R}$ represents the mean and variance parameters of the n_k kernels. The conditional density of the energy prices is thus expressed as:

$$p(\mathbf{y}|\mathbf{x}) = \sum_{k=1}^{n_k} \alpha_k(\mathbf{x})\phi_k(\mathbf{t}|\mathbf{x}), \quad \text{with } \sum_{k=1}^{n_k} \alpha_k(\mathbf{x}) = 1 \quad (4)$$

where $\alpha_k(\mathbf{x}) \in \mathbb{R}$ are the mixing coefficients, functions of the inputs through the network, combining kernel outputs into the overall distribution.

The functional mapping of input data into mixture parameters must be defined. In this work, we exploited a recurrent neural network based conditioning. The rationale behind such decision is twofold. On the one hand, as opposed to the static nonlinear mapping on a predefined window, RNNs perform the same learning task across the input sequence by weight sharing, extracting patterns on different positions. On the other hand, RNNs support the implementation of lossy summaries, forcing structuring representations in the latent state from arbitrary long input sequences. The state size becomes a hyper-parameters, tuned by analyzing the consequent effect on prediction accuracy.

Basic RNNs often result difficult to be trained, suffering the vanishing/exploding gradient problem, which results from the recurrent application of nonlinear activations [15]. Several extensions of the basic RNN unit have been proposed to address this issue. Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are the most used in practical applications nowadays. Compared to LSTMs, GRUs employ a single gating unit to control the state update and the forgetting factor [16]. It has been shown that the former provides enhanced representation power, operating as an automata with external memory, at the cost of higher complexity in terms of parameters, whereas the latter behaves more as a finite state machine [17]. In this work, we implement an RNN based on GRU cells that are well fitted with the characteristics of the problem at hand while computationally cheaper than LSTM. Rcurrence introduces a partial depth into the networks; indeed, some computational are characterized by element-wise nonlinear activations on linear input transformations [18]. Such shallow paths limits the identification of complex nonlinear mappings. Consequently, we exploited a stacked GRU architecture, aimed to extract patterns at different time scales through flexible latent representations.

A further characteristic of the proposed architecture is constituted by the input feature selection mechanisms. The identification of the subset of features to be provided to the network (including specific lags) from the available data series is often performed by time consuming trial-error procedures, leveraging on experts knowledge [19]. Automated selection results fundamental in context lacking specialized skills, as e.g., in the industrial demand side. To this end, we included an L1-norm based shrinkage factor to the input data weights of the network. As opposed to conventional L2-norm based regularizers contracting parameters to small values, sparse solutions are fostered via L1-norm, thus introducing de-facto a selection mechanism across the lags of the multi-input series. The specific penalty must be properly tuned. To such an aim, we adopted cross-validation.

The developed network architecture is formalized as follows, where we introduce a single recurrent layer to simplify notation:

$$\begin{aligned}
z_t =& sigm(W_z x_t + U_z h_{t-1} + b_z)\\
r_t =& sigm(W_r x_t + U_r h_{t-1} + b_r)\\
h_t =& (1 - z_t) \odot \tanh(W_h x_t + U_h(r_t \odot h_{t-1} + b_h))\\
&+ z_t \odot h_{t-1}\\
\mu_k =& h_t^{\mu_k}\\
\alpha_k =& \frac{\exp\left(h_t^{\alpha_k}\right)}{\sum_{j=1}^{n_k} \exp\left(h_t^{\alpha_j}\right)}\\
\sigma_k =& \ln\left(1 + \exp\left(h_t^{\sigma_k}\right)\right)
\end{aligned} \quad (5)$$

where z_t defines the update gate, r_t the reset gate, $\odot$ the Hadamard product, $W_z, W_r, W_h \in \mathbb{R}^{n_h \times n_u}$, $U_z, U_r, U_h \in \mathbb{R}^{n_h \times n_h}$, the weight matrices and $b_z, b_r, b_h \in \mathbb{R}^{n_h}$, the bias vectors. The gates include an element-wise sigmoid activation, $sigm(z) = \frac{1}{1+e^{-z}}$, while an hyperbolic tangent activation $\tanh(x) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$, is used for the hidden state equation. $h_t^{\alpha_k}$,$h_t^{\mu_k}$,$h_t^{\sigma_k}$ represent hidden state components related to specific kernel parameters. Kernel mean parameters are mapped to network outputs by adopting the uninformative prior approach. The outputs related to kernel variances are transformed by an SmoothReLU function to achieve proper variance values (i.e., positive), and to avoid convergence to solutions including null values. Mixing coefficients are processed by a softmax operation to achieve proper probability distributions. We trained the overall network by an end-to-end approach, as detailed in the following section.

B. Network training method and predicted density analysis

To train the Probabilistic EPF network, we developed the following objective function:

$$Obj = \mathcal{L} - \gamma \sum_{k=1}^{n_k} \alpha_k \ln(\alpha_k) + \delta \sum_{j=1}^{n_{W_{in}}} |\mathbf{W}_j| \quad (6)$$

The first term represents the negative log-likelihood including the Gaussian Mixture layer, that yields:

$$\mathcal{L} = -\sum_{s=1}^{S} \ln(p(\mathbf{y}^{(\mathbf{s})}|\mathbf{x}^{(\mathbf{s})})) = \sum_{s=1}^{S} \mathcal{L}^{(\mathbf{s})} \quad (7)$$

$$\mathcal{L}^{(\mathbf{s})} = -\ln\left(\sum_{k=1}^{n_k} \alpha_k\left(\mathbf{x}^{(\mathbf{s})}\right) \phi_k\left(\mathbf{y}^{(\mathbf{s})}|\mathbf{x}^{(\mathbf{s})}\right)\right) \quad (8)$$

The sample-wise gradients of the log-likelihood with reference to the mixture parameters are calculated as:

$$\frac{\partial \mathcal{L}^{(\mathbf{s})}}{\partial h^{\mu_k}} = -\pi_k\left(\mathbf{x}^{(\mathbf{s})}\right)\frac{\mu_k\left(\mathbf{x}^{(\mathbf{s})}\right) - y^{(\mathbf{s})}}{\sigma_k\left(\mathbf{x}^{(\mathbf{s})}\right)^2} \tag{9}$$

$$\frac{\partial \mathcal{L}^{(\mathbf{s})}}{\partial h^{\sigma_k}} = -\pi_k\left(\mathbf{x}^{(\mathbf{s})}\right)\left(\frac{\left\|\mathbf{y}^{(\mathbf{s})} - \mu_k\left(\mathbf{x}^{(\mathbf{s})}\right)\right\|^2}{\sigma_k\left(\mathbf{x}^{(\mathbf{s})}\right)^2} - n_y\right) \tag{10}$$

$$\frac{\partial \mathcal{L}^{(\mathbf{s})}}{\partial h^{\alpha_k}} = \alpha_k\left(\mathbf{x}^{(\mathbf{s})}\right) - \pi_k\left(\mathbf{x}^{(\mathbf{s})}\right) \tag{11}$$

where $\pi_k \in \mathbb{R}^{n_k}$ represents the posterior probabilities of the mixture components, often referred to as responsabilities, obtained by Bayes theorem from the prior probabilities α_k. Posterior probabilities are obtained for each mixture component as follows:

$$\pi_k = \frac{\alpha_k \phi_k}{\sum_{j=1}^{n_k} \alpha_j \phi_j}, \qquad \text{with} \sum_{k=1}^{n_k} \pi_k(\mathbf{x}) = 1 \tag{12}$$

The second term weights the L1-norm on the RNN input parameters, dedicated to features selection.
We introduced the third term as a regularizer on the size of the mixture by the priors entropy, increasing when mixing parameters tend to attain equivalent values [20].

The overall network is trained end-to-end by a time series cross validation approach. To this end, we adopted the Adam algorithm, conceived to tackle noisy and sparse gradients [21]. Afterwards, the trained network predicts the approximated density function of the hourly prices, conditioned on the specific values of the input sequences. From such overall description of the data generator, specific analysis can be performed. First of all, the distribution moments can be calculated. For instance, mean and variance are calculated as:

$$\hat{\mathbf{y}}(\mathbf{x}) = \sum_{k}^{n_k} \alpha_k(\mathbf{x})\boldsymbol{\mu}_k(\mathbf{x}) \tag{13}$$

$$\hat{\sigma}^2(\mathbf{x}) = \sum_{k}^{n_k} \alpha_k(\mathbf{x})\left\{\sigma_k^2(\mathbf{x}) + \left\|\mu_k(\mathbf{x}) - \sum_{j}^{n_k}\alpha_j(x)\mu_j(\mathbf{x})\right\|^2\right\} \tag{14}$$

It is worth noting that the variance is determined as a function of the input, providing a deeper characterization than conventional sum of squares-based approaches. Moreover, specific mixture components characteristics can be investigated, e.g., in terms of the related probability mass.
The extraction of the price values with higher density is typically valuable in practical application. Both iterative nonlinear optimization methods and approximated approaches can be exploited for such purpose [13]. In this work we adopted the latter, as it is computationally cheaper and faster during prediction. In particular, we approximate the most likely values by the mean of the mixture components with the larger weight, as:

$$\mu_{k^*}(\mathbf{x}) \quad \text{with} \quad k^* = \underset{k}{argmax}(\alpha_k(\mathbf{x})) \tag{15}$$

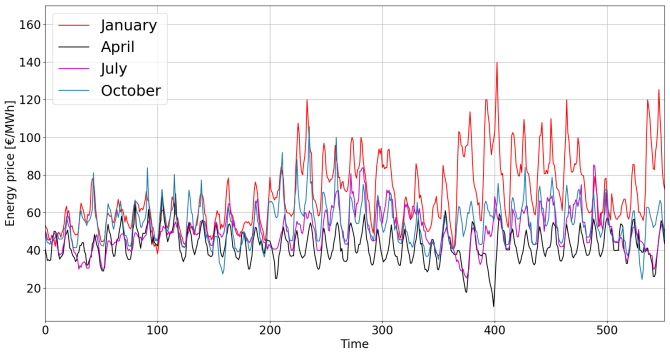

Fig. 2. Italian day-ahead market price over different months of 2017

The investigation of further statistics, e.g., to support specific application requirements, is left for future extension of the present work. Finally, the EPF model has been designed to provide multi-period forecasts (i.e., 24 hours of the next day) by employing only input features available in real bidding conditions (i.e., measures and predictors accessible on the morning of the day before). Hence, the network is expected to identify correlations and patterns both in hourly and cross-hour contexts during training.

III. Results and Discussion

We investigated the proposed probabilistic EPF approach by the application to the Italian day-ahead energy market. The dataset has been obtained from the open repositories provided by GME and Terna websites [22], including samples starting from January-2015 till the end of October-2018. The employed set of conditioning variables is composed of the overall electricity demand, the foreseen generation and the solar/wind power plants contributions. The major characteristics of the time series to be predicted are shown in Figure 2, reporting price fluctuations across different months of the same year. The observable coupled-peak structure is correlated to the major electricity load demands, typically occurring in early morning and late afternoon, characterized by season specific locations and volatility. A clearer view of the hour specific distributions of the energy price is given by the histograms in Figure 3. Besides, sensible shifts occur between working days and holidays, mostly due to different consumption patterns. A more detailed analysis of the Italian day-ahead market dataset, including descriptive statistics, can be found in [7].

We devoted the last year of the available data (i.e., from 2017/11/1 to 2018/10/31) to test set in order to investigate the performance of the EPF models across different seasonal conditions. Such subset is left to one-shot out-of-sample predictions to support a fair calculation of the performance indicators, whereas hyper-parameters tuning is tackled by a k-fold time series cross-validation (kFTsCv) approach. To this end, batches of ordered sub-sequences (including both past price values and related conditioning variables) are built by sliding a window with configurable width throughout the overall sequences. Considering the analysis performed in [7],

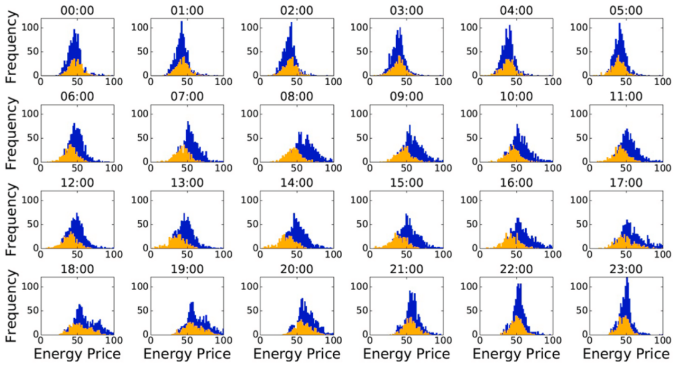

Fig. 3. Hourly price Distributions. Working(Bl) and non-working(Or) days

reporting major auto-correlations on first 24 lags, we set the width to 24. The regressors set of the feed-forward neural network and the Back Propagation Through Time length of the recurrent neural network have been configured accordingly. Subsequently, validation subsets of previous folds are included within the training set of following folds, thus increasing the amount of data used to shape model parameters and providing the latest patters of the generating process and related conditioning variables to the predictor. In detail, we employed 5 folds and a mini-batch size of 32 samples, balancing gradient estimation accuracy and related computational cost. Besides, we standardized the input series to zero mean and unit variance and interpolated daylight saving related samples.

To perform quantitative analysis of models predictions, we employed Symmetric Mean Absolute Percentage Error (sMAPE) and Continuous Ranked Probability Score (CRPS):

$$sMAPE = \frac{100}{S}\sum_{s=1}^{S}\frac{\left|\hat{y}^{(s)} - y^{(s)}\right|}{\left(\left|\hat{y}^{(s)}\right| + \left|y^{(s)}\right|\right)/2} \tag{16}$$

$$CRPS = \frac{1}{MS}\left[\sum_{s=1}^{S}\sum_{j=1}^{M}\left|\hat{y}^{(s)} - y^{(s)}\right| - \frac{1}{2}\sum_{s=1}^{S}\sum_{j=1}^{M}\left|\hat{y}^{(s)} - \hat{y}'^{(s)}\right|\right] \tag{17}$$

where $\hat{y}$ represent the predicted price, y the target value, M a set of independent probabilistic EPF model samples. sMAPE provides a scale independent view of the forecast performance, reducing conventional MAPE sensitity to small values. CRPS, summarizing calibration (i.e., forecast error) and sharpness (i.e., distribution concentration) has been employed following the indication of [1] on probabilistic EPF models analysis.

We developed the neural networks by means of Tensorflow 2.0, including the Tensorflow Probability library providing facilities to develop probabilistic network layers. The Gaussian Mixture component has been implemented through a custom Keras layer, by coding specialized 'init', 'call' and 'loss' functions. The hyperparameter set includes the layers, the units in each layer, epochs and related early stop patience (i.e., interrupt training loop when prediction accuracy stop decreasing), and objective function penalties δ, γ. By kFTsCv, we identified a configuration of 20 epochs with a patience of 30, $\delta = 0.01$, $\gamma = 0.02$, and a network architecture with 2 layers of 50 GRU cells to perform test set experiments. We did not measure sensible performance increases during tests of larger architectures, while still impacting on computational costs. Table 1 reports the results achieved on the test set.

The proposed EPF approach has been compared to the Bayesian Neural Network (BNN) based method proposed in [7] and to a deterministic RNN based model. We considered the former since it represents, to the best of author knowledge, the unique probabilistic EPF method exploiting a Neural Network in the literature. The latter EPF model is constituted by the same network configuration of the MDN-RNN, replacing the GMM layer with a conventional linear layer trained by sum of square error. It has been included within the experimental set-up in order to investigate the contribution within the model of the defined recurrent network architecture, not discussed within the literature. For the BNN, we maintained the configuration of the original paper. We did not compute CRPS for the deterministic RNN due to the infeasible sampling.

Notably, the developed RNN-based model provides a relevant increase of forecast performances, as compared to the FFNN based architecture. Indeed, it is worth noting that the BNN was compared to a deterministic feedforward neural network in [7], achieving consistent results. The MDN-RNN obtained also lower prediction errors than the RNN. Perhaps, such result is related to the specific characteristics of the EPF problem under treatment (e.g., actual distribution form vs Gaussian assump-

TABLE I
HOURLY PRICE PREDICTION PERFORMANCES

	00	***01***	***02***	***03***	***04***	***05***
sMAPE Bayes-NN	9.3	10.2	8.9	11.1	10.8	12.4
sMAPE Deter-RNN	9.3	10.7	10.6	10.3	12.2	10.1
sMAPE MDN-RNN	8.0	8.9	9.6	11.1	12.2	10.4
CRPS Bayes-NN	4.6	4.7	4.8	5.3	5.2	4.8
CRPS MDN-RNN	3.8	3.9	3.9	4.3	4.9	4.5
	06	***07***	***08***	***09***	***10***	***11***
sMAPE Bayes-NN	10.2	13.0	9.3	11.8	9.8	12.8
sMAPE Deter-RNN	9.6	9.5	11.0	10.1	9.3	9.5
sMAPE MDN-RNN	10.0	10.4	10.9	10.0	9.3	9.5
CRPS Bayes-NN	5.3	6.6	8.3	7.5	6.5	6.5
CRPS MDN-RNN	5.0	5.7	6.7	6.0	5.2	5.1
	12	***13***	***14***	***15***	***16***	***17***
sMAPE Bayes-NN	13.2	11.8	11.3	11.7	13.4	10.9
sMAPE Deter-RNN	9.5	10.6	12.0	11.8	11.1	11.2
sMAPE MDN-RNN	9.7	10.2	11.6	12.0	11.4	10.8
CRPS Bayes-NN	6.0	6.2	6.2	7.2	8.0	7.4
CRPS MDN-RNN	4.8	4.7	5.4	6.0	6.4	6.5
	18	***19***	***20***	***21***	***22***	***23***
sMAPE Bayes-NN	12.1	11.9	10.8	13.8	12.1	13.1
sMAPE Deter-RNN	9.5	9.5	9.0	12.6	8.1	8.9
sMAPE MDN-RNN	8.9	9.6	8.2	7.3	7.9	8.3
CRPS Bayes-NN	8.2	7.5	7.2	6.1	5.5	4.7
CRPS MDN-RNN	5.9	6.7	5.3	4.4	4.3	4.1

	Bayes-NN	***Deter-RNN***	***MDN-RNN***
sMAPE	0.115	0.102	0.098
CRPS	7.46	-	5.15

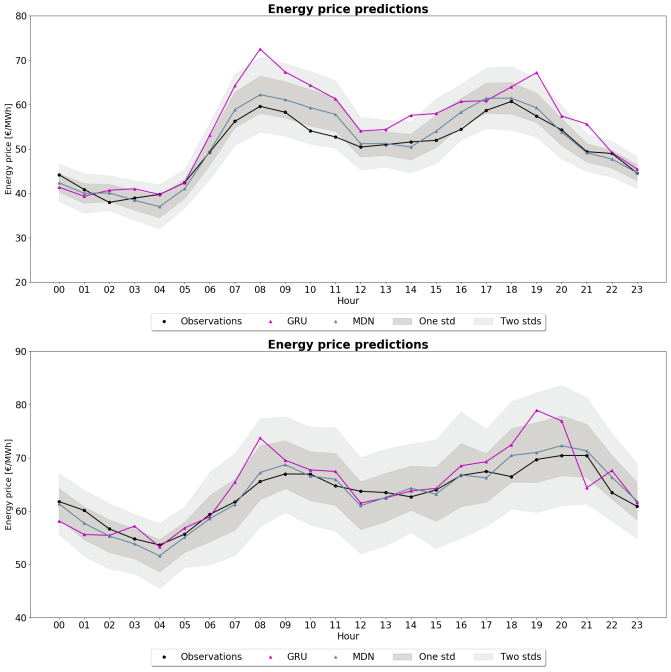

Fig. 4. Samples of Price forecasts

tion, etc.). Further increases of performances are expected by application to markets characterized by more complex dynamics. Still, the major benefit of the proposed approach, as compared to the deterministic RNN, is highlighted by Figure 4, including point forecasts as well as prediction/hour specific standard deviations. Remarkably, the probabilistic formulation extends conventional point forecast techniques by providing uncertainty indications, representing a fundamental tool for the user to achieve more informative decision making. For instance, enhanced what-if multi-scenario analysis can be performed as well as detailed assessment of the robustness of the energy-aware production scheduling strategy [23].

IV. Conclusion and Next Steps

In this work we have presented a novel probabilistic energy price forecast method enabling the identification of general conditional distributions, extending previously proposed neural model based on the Gaussian assumption. To such an aim, we develop a EPF model including a Gaussian Mixture layer, parametrized by a Recurrent Neural Network with Gated Recurrent Units, processing conditioning variables sequences including past values of the hourly price. Then, we developed an architecture performing multi-hour predictions, conceived to learn patterns both in hourly and cross-hour contexts. Moreover, we included an L1 norm based input feature selection mechanisms within the input layer of the network, aimed to identify the most informative subset across the lags of the multi-input series. By application to a real price market dataset, we showed the capability of the developed network to achieve increased forecast accuracy. Compared to state of the art point forecast techniques exploiting Neural Networks, the proposed method provides explicit forecast uncertainty indications to the users, thus enabling more informative decision making and enhanced energy-aware optimization strategies. Next developments will include the integration of further exogenous variables within the input set (e.g, prices of connected regional markets, etc.), the application to other energy markets (e.g., NordPool) and the investigation of alternative network configurations.

References

[1] J. Nowotarski and R. Weron, "Recent advances in electricity price forecasting: A review of probabilistic forecasting," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1548 – 1568, 2018.

[2] B. Uniejewski, J. Nowotarski, and R. Weron, "Automated variable selection and shrinkage for day-ahead electricity price forecasting," HSC Research Reports HSC/16/06, Hugo Steinhaus Center, Wroclaw University of Technology, July 2016.

[3] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International Journal of Forecasting*, vol. 30, no. 4, pp. 1030 – 1081, 2014.

[4] D. Keles, J. Scelle, F. Paraschiv, and W. Fichtner, "Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks," *Applied Energy*, vol. 162, pp. 218 – 230, 2016.

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

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

[7] A. Brusaferri, M. Matteucci, P. Portolani, and A. Vitali, "Bayesian deep learning based method for probabilistic forecast of day-ahead electricity prices," *Applied Energy*, vol. 250, pp. 1158 – 1175, 2019.

[8] C. M. Bishop, "Mixture density networks," research report, Aston University, Neural Computing Research Group, 1994.

[9] A. Graves, "Generating sequences with recurrent neural networks.," *CoRR*, vol. abs/1308.0850, 2013.

[10] O. Makansi, E. Ilg, O. Cicek, and T. Brox, "Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction," pp. 7137–7146, 06 2019.

[11] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Advances in Neural Information Processing Systems 31* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), pp. 2450–2462, Curran Associates, Inc., 2018.

[12] L. U. Hjorth and I. T. Nabney, "Regularisation of mixture density networks," in *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, pp. 521–526 vol.2, 1999.

[13] C. Rupprecht, I. Laina, R. Dipietro, M. Baust, F. Tombari, N. Navab, and G. Hager, "Learning in an uncertain world: Representing ambiguity through multiple hypotheses," vol. 2017-October, 2017.

[14] G. Mclachlan and K. Basford, *Mixture Models: Inference and Applications to Clustering*, vol. 38. 01 1988.

[15] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, vol. 385. 01 2012.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT, 2016.

[17] C. Wang and M. Niepert, "State-Regularized Recurrent Neural Networks," *arXiv e-prints*, p. arXiv:1901.08817, Jan 2019.

[18] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks, arxiv:1312.6026," 12 2013.

[19] A. Brusaferri, L. Fagiano, M. Matteucci, and A. Vitali, "Day ahead electricity price forecast by narx model with lasso based features selection," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, pp. 1051–1056, July 2019.

[20] S. Das and M. Mozer, "Dynamic on-line clustering and state extraction: An approach to symbolic learning," *Neural Networks*, vol. 11, no. 1, pp. 53 – 64, 1998.

[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[22] "Gestione mercato elettrico." http://www.mercatoelettrico.org.

[23] D. Ramin, S. Spinelli, and A. Brusaferri, "Demand-side management via optimal production scheduling in power-intensive industries: The case of metal casting process," *Applied Energy*, vol. 225, pp. 622 – 636, 2018.