Bayesian deep learning based method for probabilistic forecast of day-ahead electricity prices

Alessandro Brusaferri[a,b,*] Matteo Matteucci[b], Pietro Portolani[a], Andrea Vitali[a]

[a] *CNR, Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, via A.Corti 12, Milan, Italy*
[b] *Politecnico di Milano, Department of Electronics, Informatics and Bioengineering, via Ponzio 34/5, Milan , Italy*

Abstract

The availability of accurate day-ahead energy prices forecasts is crucial to achieve a successful participation to liberalized electricity markets. Moreover, forecasting systems providing prediction intervals and densities (i.e. probabilistic forecasting) are fundamental to enable enhanced bidding and planning strategies considering uncertainty explicitly. Nonetheless, the vast majority of available approaches focus on point forecast. Therefore, we propose a novel methodology for probabilistic energy price forecast based on Bayesian deep learning techniques. A specific training method has been deployed to guarantee scalability to complex network architectures. Moreover, we developed a model originally supporting heteroscedasticity, thus avoiding the common homoscedastic assumption with related preprocessing effort. Experiments have been performed on two day-ahead markets characterized by different behaviors. Then, we demonstrated the capability of the proposed method to achieve robust performances in out-of-sample conditions while providing forecast uncertainty indications.

Keywords: Electricity Price forecasting; Probabilistic forecasting; Deep learning; Bayesian Learning; Neural Network

1. Introduction

Since the last two decades, many countries worldwide performed the transition from traditional government-controlled to competitive electricity markets [1]. Within a deregulated environment, day-ahead trading is initiated by producers/consumers submitting bidding proposals to the market operator for selling/buying energy blocks. Afterwards, the market operator applies specific strategies for clearing the market while guaranteeing proper balancing and respect of the transmission network constraints, resulting in a market price and acknowledged selling/buying proposals [2]. Consequently, a strong competition emerged between an increasing number of market participants trying to exploit new opportunities by reducing costs and risks while increasing margins [3], [4], [5].

In this context, a robust day-ahead energy price forecast (EPF) represents a key strategic tool for both utilities, retailers, aggregators and consumers to succeed within the liberalized framework by performing effective bidding strategies. On one hand, purchasers could properly reshape consumption patterns trying to target lower price profiles. Indeed, energy-intensive companies are increasingly integrating day-ahead prices forecasting within decision-making strategies [6]. On the other hand, producers need proper prices predictions within simulation and optimization tools for generation asset management, so to perform accurate costs/benefits estimates and adjust production plans and bids accordingly [7], [8], [9].

* Corresponding author
Email addresses: alessandro.brusaferri@stiima.cnr.it (A. Brusaferri), matteo.matteucci@polimi.it (M. Matteucci), pietro.portolani@stiima.cnr.it (P. Portolani), andrea.vitali@stiima.cnr.it (A. Vitali)

Published Journal Article available at: http://dx.doi.org/10.1016%2Fj.apenergy.2019.05.068

Besides, prices volatility is strictly related to stable grid balancing. Indeed, precise forecasting indirectly contributes to overall system stability by supporting the creation of a converging win-win situation between the system operator and market participants [10], [11].

At current time, day-ahead forecast constitutes a fundamental ingredient for further research and technological developments areas such as optimal commitment of generation units, energy-aware production scheduling, and power system simulation [12].

Unluckily, achieving a highly accurate prediction is quite challenging due to several issues. Compared to other types of commodities, electricity is still not economically storable and a closed equilibrium between production and consumption is mandatory for power network stability [13], [14], [15]. Besides, both consumption and production profiles strongly depend on multiple exogenous variables such as weather conditions (e.g. temperature, wind, solar radiations, etc.), human activities and business related consumption patterns (e.g. weekdays, holidays, peak hours, etc.), and fuel price fluctuations [16]. Indeed, it has been shown that the power markets exhibit a very specific behavior which is quite uncommon compared to other commodities. Typical characteristics include strong-nonlinearity, non-constant mean and variance, significant short-term peaks, daily/weekly seasonality and higher volatility, usually two orders of magnitude higher than other financial trading or utilities [3], [17],[18],[19]. Moreover, the extending exploitation of power generation systems from renewable sources represent a further complexity to be properly tackled [20]. Indeed, technologies as wind and solar power plants are characterized by higher volatility and short-time fluctuations, with consequent uncertainties on effective hourly available power.

A lot of scientific effort is being dedicated to develop enhanced techniques for day-ahead energy prices forecast, representing a highly active and continuously improving field of research [21], [22], [3].

1.1. Literature review

Several methods have been proposed within the quite wide research literature on this topic, each with specific strengths and weaknesses, typically emerged as application specific. Different classification approaches have been followed in survey papers. Here we adopt the mostly considered arrangement of [3] organized into five groups, namely fundamental, reduced-form, multi-agent, statistical, and computational intelligence models.

Fundamental models exploit explicit formulations of the backbone relationship between the major drivers of electricity trading, including a detailed characterization of demand and generation dynamics, e.g. [23], [24], [25], [26]. Despite their potential capabilities, practical applications are limited by the lack of information accessible to properly set-up the models. Indeed, plant characteristics and costs, as well as detailed transmission capabilities, are often available as weekly/monthly aggregated data [27]. Therefore, a lot of effort is usually required to properly collect and structure needed technical information and run-time data. Furthermore, predictions are strongly affected by specific assumptions on functional associations and stochastic behavior of integrated drivers. Consequently, fundamental models are often more effective for medium terms forecast and risk management than for short-term hourly forecast [3].

Reduced-form models directly formulate the dynamics of spot price. To this end, Jump-diffusion [28] and Markov regime-switching-based techniques are often adopted [29]. Several research studies displayed their capabilities in replicating major dynamics at daily level as well as price spikes predictions. On the other hand, limited performances have been reported on day-ahead hourly price forecast problems [30], [31].

Multi-agent approaches emerged as extensions of traditional cost-based models to properly cope with competitive dynamics [32]. Indeed, latter methods were conceived for stable regulated markets characterized by low uncertainty [33]. To achieve accurate predictions major market components have to be properly deployed, such as the list of market players, adopted strategies by heterogeneous agents, multi-agent interaction mechanisms, etc. Unfortunately, most of the required information is

typically not available to the forecaster. Therefore, assumptions are often introduced leading to potential modelling inaccuracy. Furthermore, a sensible effort is required to set-up highly granular models suitable for short-term predictions. In fact, multi-agent systems tend to be more feasible to support qualitative analysis than accurate daily predictions. Most applications targeted exploration of major market dynamics and strategy assessment [34]. Besides, previous studies considered their integration as sub-components within hybrid forecast systems [35].

Statistical and computational intelligence represent EPF techniques most exploited nowadays, mainly due to their capabilities in supporting short-term forecasts without requiring detailed systems modelling.

In details, statistical methods class comprises Auto-Regressive Moving Average models with eXogenous input (ARMAX) and related subclasses (e.g. AR, ARX, ARMA) and extensions, aimed at performing prices predictions by identifying the time correlations between the sequences of energy prices and explanatory input variables (e.g. weather, demand, etc.) [36]. The Auto-Regressive counterpart explores the relations between a specific time value and past lags of the price series. On the other hand, Moving Average terms are meant to identify the dynamics of the stochastic process characterizing the residuals. In the classical form, linear multi-variate additive or multiplicative models are usually adopted. Notably, ARMAX modelling assumes a weakly stationary time series. Therefore, a proper transformation of EPF time series into stationary form has to be performed, (e.g. by differencing). Alternatively, extensions to the classical models can be adopted, as ARIMAX model including differencing by an integrating term or Seasonal-ARIMAX including specific seasonal patterns not manageable by lag-1 differencing, as for example on hourly, daily or weekly periodicity [37]. Actually, ARMAX-based techniques have been explored extensively for energy prices prediction, (see e.g. [38], [39], [40]). Furthermore, Generalized Auto-Regressive Conditional Heteroskedastic models have been investigated to address the limiting homoscedasticity assumption of straight ARMAX models. In this case, conditional variances of the time series are represented by weighted sums of squared past observations. Hence, hybrid approaches are often adopted, combining ARMAX-GARCH-based techniques, where GARCH typically works on the residual [41], [42].

In general, recognized major strength of statistical models resides in their simple interpretation by users whereas major limitation regards incapability to deal with complex non-linear relations within the multi-variate data sequences, as in the case of energy-prices forecast. To cope with such limitation, nonlinear extensions have been investigated, as ARX models integrating a nonlinear component (i.e. NARX) [43]. The nonlinear term is usually expressed by predefined functional relationships, parametric functions (e.g. polynomial) or by crossing the border toward computational intelligence techniques, integrating for example Kernel Methods or Neural Network within ARX components [44], [45].

In wider terms, computational intelligence techniques involve an extensive spectrum of methods, including neural networks, kernel methods, support vector machines, fuzzy logic, genetic algorithms, swarm intelligence (see e.g. review in [3]). In this context, neural network based techniques are often exploited mainly due to their flexibility and capability to represent complex nonlinearities [20],[46],[47],[1]. Notably, deep neural network architectures provide capabilities to learn hierarchical features from the data set while providing a more efficient representation than shallow models and improving generalization [57]. Several scientific results support such hypothesis, including both theoretical (e.g. [60], [61], [58]) and empirical studies (e.g. [62], [63], [64]). Indeed, deep networks are nowadays achieving state of the art results on complex machine learning tasks (e.g. computer vision, natural language processing, speech recognition etc.).

From the architectural point of view, two major classes of neural networks exist, namely feedforward and recurrent. The former implements a static mapping between input and output variables in the data set, which are typically considered as independent and identically distributed (i.e. i.i.d.). Therefore, predictions are performed independently from previously seen inputs. The latter implements feedback loops within the network enabling the capability to discover relations among ordered sequences of input and output data. Notably, the vast majority of available literature on day-

ahead energy prices forecast exploits feedforward neural networks, as in [48], [49],[50], [51], [52], [53], [54], [55]. Modern deep and recurrent architectures (e.g. Long short-term memory) were not investigated until very recently (see e.g. [56],[65],[108],[109],[110]). In [65], authors performed a broad empirical investigation of alternative methods for EPF, demonstrating state of the art performances of deep neural networks compared to alternative methods.

Regardless of the specific method employed, model averaging techniques (also referred to as ensemble) have been investigated to improve EPF accuracy by combining predictions from multiple models (see e.g. [111],[112]). In particular, the comprehensive empirical study performed in [112] reported superior performance under market conditions characterized by low volatility.

Notably, the vast majority of available methodologies, including studies considering deep neural networks, focus on point forecasts. Nevertheless, practitioners would strongly benefit from forecasting systems providing predictions intervals and densities (i.e. probabilistic forecasting). Indeed, according to the recent extensive review performed by [66], probabilistic forecast provides several appealing facilities, including improved assessment of future uncertainty, risk evaluation and ability to plan multiple strategies for the range of possible prices outcomes. As remarked in [104], probabilistic forecast represents a valuable tool for generation companies (e.g. multi-scenario operational planning and trading) as well as large electrical consumers to foster enhanced participation to the market. Considering energy-intensive process industries for example (e.g., Iron&Steel manufacturing, Chemical plant utilities), multiple production schedules (see e.g., [105],[11],[113]) can be generated and compared ex-ante by sampling from the EPF distribution, thus enabling what-if scenario/consumption analysis before trading.

Currently available methods within such "fascinating but still underdeveloped" field (as stated in [66]) belong to the following families: historical simulation, distribution-based probabilistic forecast, bootstrapped Prediction Intervals and Quantile Regression Average. In this context, almost all techniques exploiting neural networks are formed on deterministic machines trained by conventional maximum likelihood based methods. EPF uncertainties are commonly addressed including zero mean Gaussian residuals with constant variance (see e.g., [67]) or bootstrap Prediction Intervals by sampled residuals (see e.g., [68]). To the best of our knowledge, the only exception to this trend is represented by [69], investigating a Bayesian method based on Markov Chain Monte Carlo. Nevertheless, the proposed technique has been deployed for a tiny shallow network with five neurons. Then, point forecasts are generated by a Monte Carlo approximation, assuming a homoscedastic time series behavior. Summarizing, to the best of our knowledge, no previous method has explored Bayesian deep learning for probabilistic EPF.

1.2. Contribution and Organization of the paper

Starting from state of the art techniques and considering reported open issues, the objective of the present study is to develop a novel methodology for probabilistic EPF exploiting Bayesian deep learning techniques. Specifically, major contributions of this paper include:

- support the development of the "fascinating but still underdeveloped" field of probabilistic EPF by extending recently proposed deep learning methods for point estimate;
- robust performances in out-of-sample forecasting conditions while providing uncertainty indications, thus enabling enhanced bidding strategies and decision-making processes;
- a neural network model originally supporting heteroscedasticity within the EPF system, thus avoiding common homoscedastic assumption with related time series preprocessing effort;
- deployed training methodology scalable to complex network architectures, thus enabling path from potential theoretical capabilities of neural networks to effective learning, while trying to simplify usage and management by industrial practitioners;
- EPF model built on posterior distributions, providing a natural regularization effect without

detailed hyperparameters tuning;

- experiments on two different energy markets, characterized by dissimilar behavior (e.g. different penetration levels of renewable sources, different climatic conditions, different demand patterns, etc.) to illustrate reusability of the proposed technique.

The rest of the paper is organized as follows. Section 2 deepens the problem at hand by analyzing the Italian energy market price (PUN, i.e. Prezzo Unico Nazionale) time series. Section 3 describes in detail the proposed methodology for probabilistic EPF. Section 4 reports the application of the proposed method to the Italian and Belgian day-ahead energy markets, reporting achieved results. Section 5 summarizes conclusion and foreseen future developments.

2. Problem description

As anticipated in the previous section, we evaluated the proposed EPF methodology on two markets to demonstrate both achieved performances and reusability in different contexts. Following subsections report the analysis of Italian PUN price since lacking within the research literature. Major characteristics are illustrated, remarking the compliance with typical behavior observed on further day-ahead markets reported within previous studies (see e.g. [3]). Details regarding Belgian energy prices could be found in [10].

2.1. Employed data set

We obtained the whole dataset via the Entso-e Transparency Platform [70] and GME/Terna websites [71]. The available time series ranges from January 2015 to November 2018. All data concerning day-ahead prices, energy load, generation, etc. are made available within the transparency platform. Afterwards, we preprocessed daylight related effects by removing extra data and interpolating missing sample. Finally, we divided the whole database into three subsets. The former two were specifically dedicated to training and validation while the latter to testing over models in out-of-sample conditions. In particular, we considered the following arrangement:

- Training set: from January 6, 2015 to October 31, 2016
- Validation set: from November 1, 2016 to October 31, 2017
- Test set: from November 1, 2017 to October 31, 2018

We chose to leave an entire year of most recent data to test set in order to explore forecast performances throughout different months characterized by specific behavior, as further detailed within the following sections. The test set has been used exclusively for the out-of-sample forecasts investigations. It is worth noting here that the adoption of a validation set is not mandatory by the deployment of straight Bayesian methods. Nevertheless, we first tuned and compared the considered models' architectures within training/validation set, in order to obtain unbiased setups. Indeed, a fair investigation must approach test data just in final "one-shot" prediction experiments.

Finally, we employed time series cross validation methods, as further detailed within Section 4.

2.2. Explanatory data analysis

We started from a straight visual inspection and descriptive statistics of the dataset to gather central tendency, a measure of dispersion, and the overall shape of the available samples distributions. As discussed in Section 1, day-ahead energy markets typically exhibit a non-stationary and quite seasonal behavior. Such general considerations are hereafter deepened, evaluating specific characteristics of the Italian market.

Figure 1 compares the shapes of spot market prices (in Euro/MWh) during months from different seasons evenly distributed throughout 2017, namely January, April, July and October.

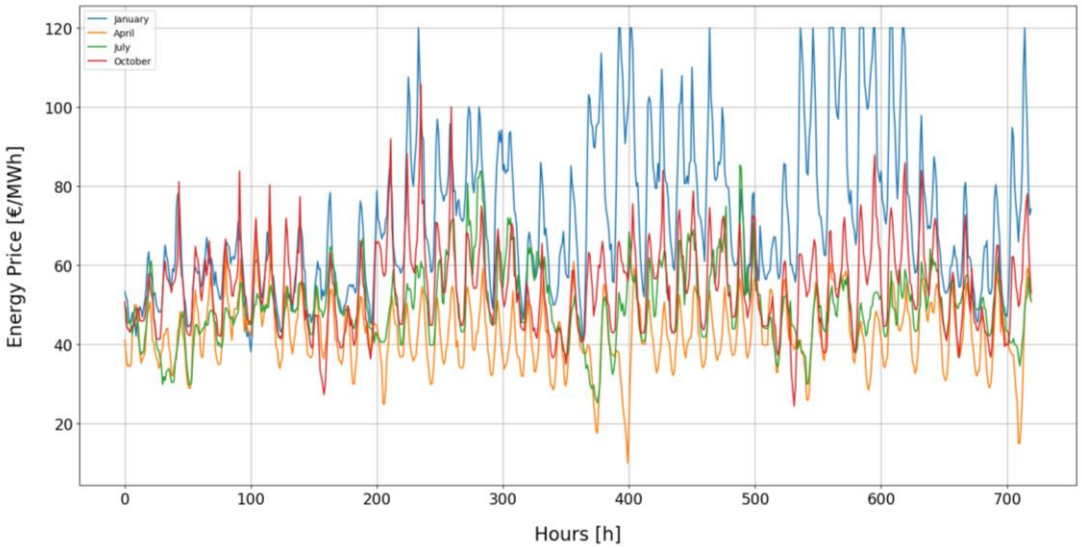

Figure 1: Day-ahead prices throughout different months of the year 2017

The figure outlines the volatile behavior of energy prices in Italy, characterized by strong variability in specific periods of the year. Moreover, the curves display a quite typical repetitive pattern, constituted of double peak shapes. Nevertheless, magnitude and extensions of the patterns are sensibly different across the seasons. January manifests the highest prices on average as well as more pronounced peaks. April appears as the series with the lowest scale while July and October are somehow in between. Moreover, the daily profile of the energy price varies considerably throughout the different seasonality cycles as introduced above. Specific daily prices curves patterns are even more evident within Figure 2, reporting evenly spaced days within the same month. Furthermore, the specific shape of nonworking days is shown, for example, January 1st with reference to January 30th.

Patterns and main structure of day-ahead time series throughout different months are illustrated also by Figure 3. In particular, the heat maps report the matrix of hourly prices on the month days. Each figure displays a clear couple of warmer vertical bands, representing peak prices. Furthermore, cooler horizontal bands are clearly visible, representing non-working days. In fact, a weekly repetitive pattern of working/nonworking day cluster is quite evident on each month.

Month-specific distributions of electricity prices are even more evident by Box and Whisker plots reported in Figure 4. Specifically, the figure depicts month-aggregated hourly prices throughout 2017, where boxes represent related upper and lower quartiles, and whisker lengths were set to 1.5 times the interquartile range. Samples considered as outliers are displayed as individual points.

Figure 5 provides an overall view on hour specific behaviors, reporting the histograms of energy prices during different periods. The samples distributions differences are quite evident, in particular between working and non-working days. Table A and Table B (reported in appendix) summarize major quantitative details by reporting related descriptive statistics.

Clearly, working days manifest a morning price peak within the time range 08:00-10:00 a.m. whereas morning prices over non-working days are more uniform. On the other hand, an afternoon peak is visible on both working and non-working day during the slot 18.00-21:00 (i.e. 06:00-09:00 p.m.). This structure might depend on load and generation ramps within such hourly slots, with a sensible shift depending on specific seasonal conditions. Evidently, such conditions are amplified during working days, characterized by a major difference between peak and off-peak loads.

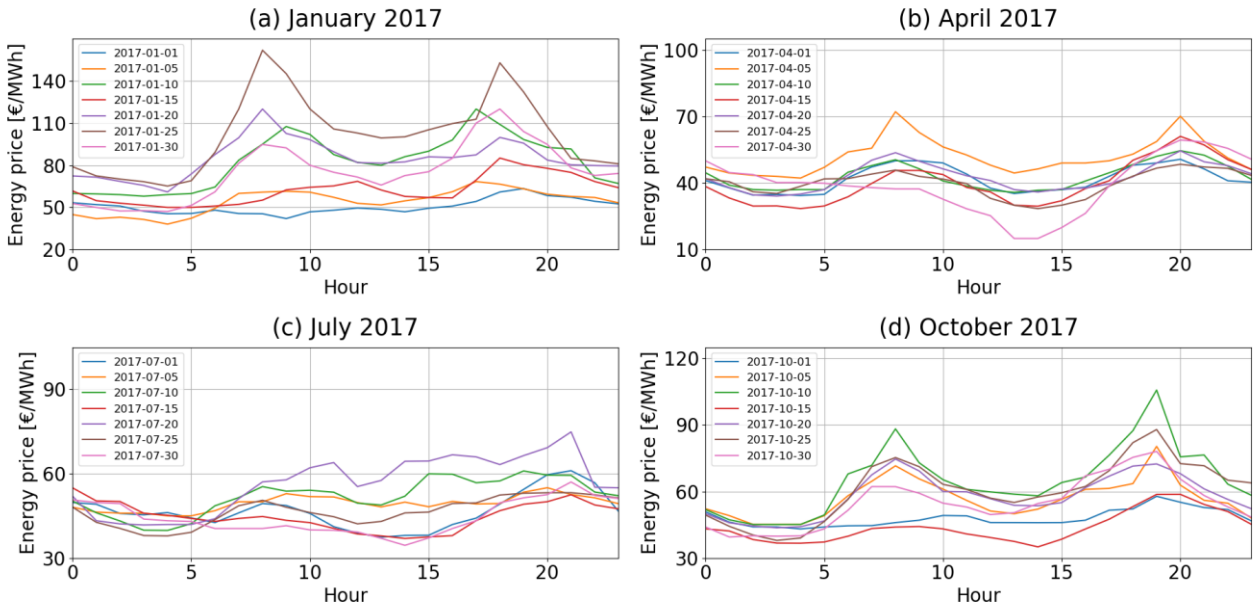

Figure 2: Day-ahead hourly prices on different days of a month:
(a) January 2017, (b) April 2017, (c) July 2017, (d) October 2017

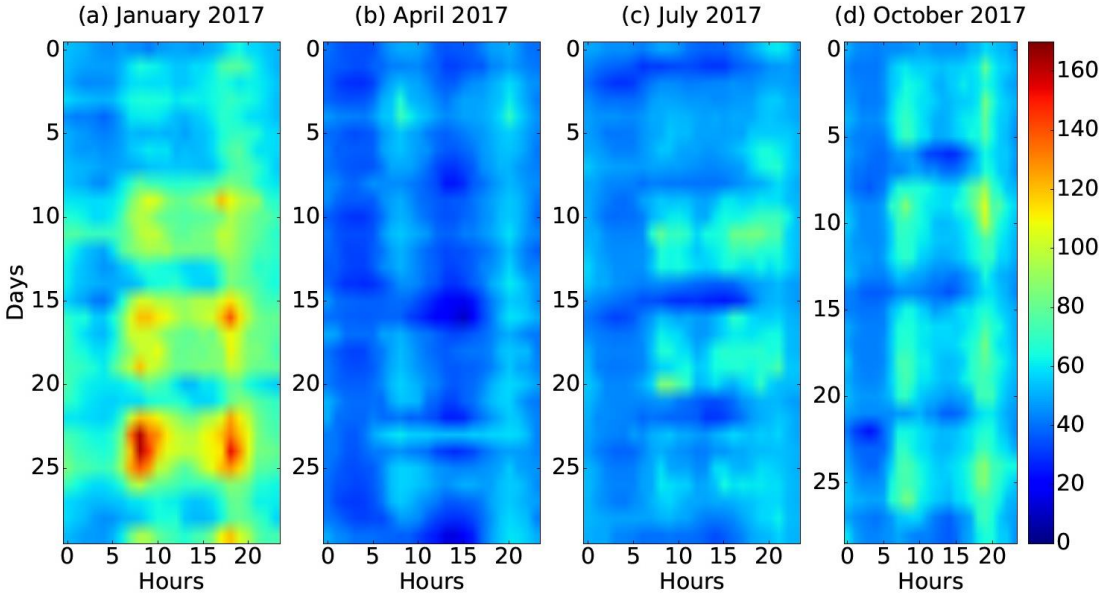

Figure 3: Heat map of Day-ahead prices on different months of 2017

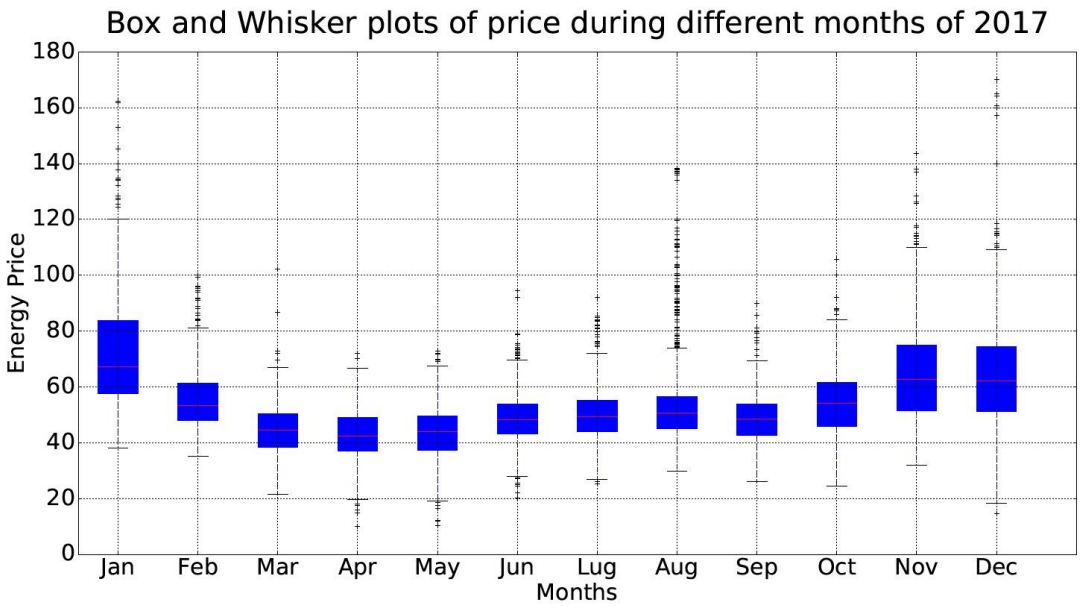

Figure 4: Box and Whisker plots of hourly price by month

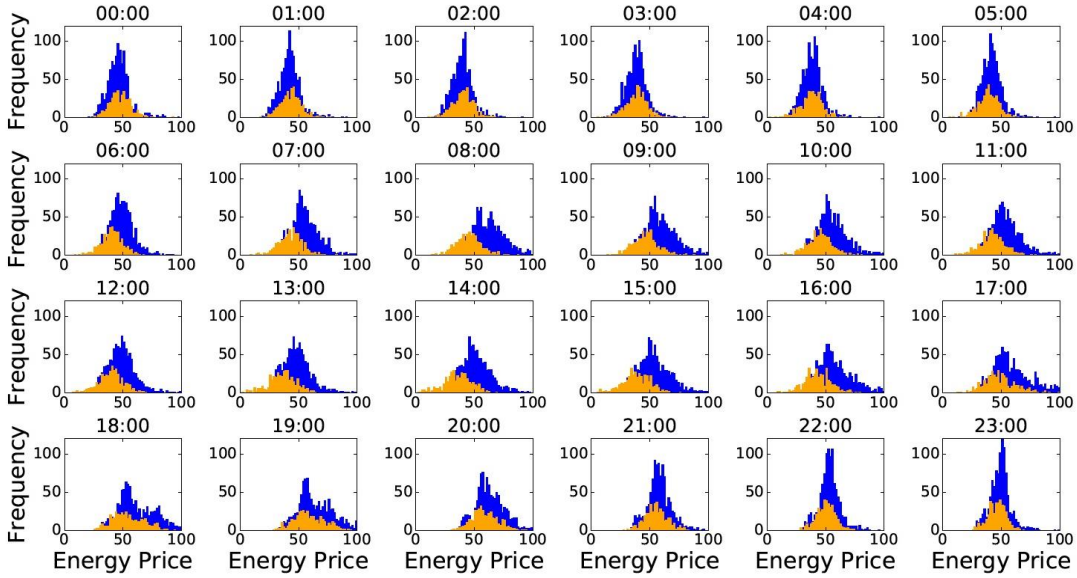

Figure 5: Samples distributions on working (blue) and non-working days (orange)

2.2.1. Seasonal unit root and partial autocorrelation

In order to investigate the presence of seasonal unit root in the Italian PUN Time Series we employed the method reported in [103], representing a generalization of the Hylleberg, Engle, Granger and Yoo (HEGY) test for any periodicity. We employed the gretl open source software package [106] for such purpose. Results are displayed in Table 1. Considering reported values of Fc, Fs and $t_{\pi 0}$, $t_{\pi_{S/2}}$, F_j^{ab} with j=1, 2, …, 11, the presence of unit roots can be rejected at a 5% level of significance on cycles with period 24, 8, 3.43, 2.67, 2.4, 2.18, 2.

Table 1. HEGY test

	Test statistic	**p-value**	**Frequency**	**Period**
$t_{\pi 0}$	-2.11	0.24880	0	∞

F_1^{ab}	6.45	0.04210	$\pm\frac{\pi}{12}$	24
F_2^{ab}	5.65	0.07018	$\pm\frac{\pi}{6}$	12
F_3^{ab}	10.50	0.00426	$\pm\frac{\pi}{4}$	8
F_4^{ab}	4.10	0.16969	$\pm\frac{\pi}{3}$	6
F_5^{ab}	3.01	0.30856	$\pm\frac{5\pi}{12}$	4.8
F_6^{ab}	4.68	0.12406	$\pm\frac{\pi}{2}$	4
F_7^{ab}	6.34	0.04549	$\pm\frac{7\pi}{12}$	3.43
F_8^{ab}	5.64	0.07039	$\pm\frac{2\pi}{3}$	3
F_9^{ab}	8.06	0.01413	$\pm\frac{3\pi}{4}$	2.67
F_{10}^{ab}	7.24	0.02410	$\pm\frac{5\pi}{6}$	2.4
F_{11}^{ab}	7.48	0.02044	$\pm\frac{11\pi}{12}$	2.18
$t_{\pi_{S/2}}$	-3.20	0.01695	π	2
F_s	6.66	0.07378	All seasonal cycles	
F_c	6.57	0.33998	All seasonal cycles + zero frequency	

We then investigated the partial autocorrelation function within price time series. Following previous studies (see e.g. [20]) we focused the analysis within the recent horizon. Figure 6 shows the first 200 lags and provides evidence about daily and weekly correlations in the Italian PUN. Indeed, cyclic spikes every 24 steps are visible. Clearly, partial autocorrelations on first 24 lags dominate the subsequent lags. It is worth noting here that a certain longer term seasonality (e.g. monthly, yearly) could be observed within energy prices time series. The detailed investigation of such potential features and their eventual integration within the presented EPF model is left to future extensions of the present work.

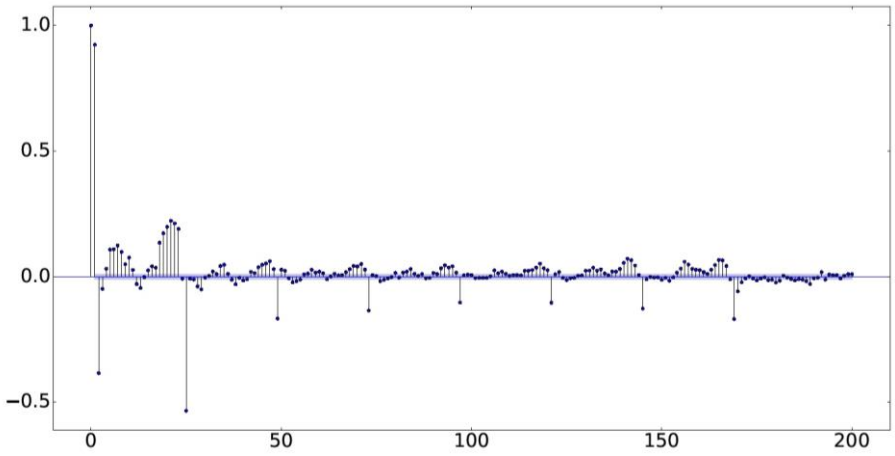

Figure 6: Day-ahead price Partial Autocorrelation Function

3. Methodology

In this section, we describe the proposed method for probabilistic EPF. To this end, we introduce theoretical foundations of Bayesian learning as well as its application to achieve probabilistic EPF,

thus providing predictive distributions and related confidences as output. Then we illustrate how we addressed heteroscedasticity within proposed forecasting system, by integrating a dedicated neural network within a parametrized variance mapping.

Subsequently, we report the major issues we encountered while setting up a practical Bayesian deep leaning system for EPF, with particular reference to the computational intractability of a pure brute-force inference approach. Therefore, we will deepen the specific technique we deployed, leveraging on variational interference. Finally, we report the neural network architectures exploited in the present study as well as the conditioning variables provided as inputs to the models.

3.1. EPF by Bayesian learning

To address the challenges reported above, we adopted Bayesian deep learning within the proposed EPF forecast method. Compared to previous neural network-based approaches for EPF, in Bayesian deep learning probabilities are explicitly included to represent uncertainties in modelling as well as random noise within the residuals.

In general terms, Bayesian training of neural networks could be addressed by treating the activations functions as stochastic variables (as performed e.g. by [72],[73]) or the weights (see e.g. [74],[75]). In the present work, we followed the latter family of methods. Such choice enabled both a mechanism to deal with uncertainties about specific parametrizations of the designed neural network form and a natural regularization effect, as described later. Therefore, deterministic parameters were replaced by probability distribution functions over the weights space, summarizing the relative prior beliefs. Practically, specific weights values are expected to be more likely before any observation is performed. As an example, zero mean Gaussian prior distributions might be introduced to provide insights and preferences towards simpler models. Such an effect is achieved by fostering most parameters around zero. Taking inspiration from [74], Figure 7 sketches the major differences between a deterministic deep neural network (i.e. a Multi Layer Perceptron in this case) and a neural network made Bayesian by stochastic weights.

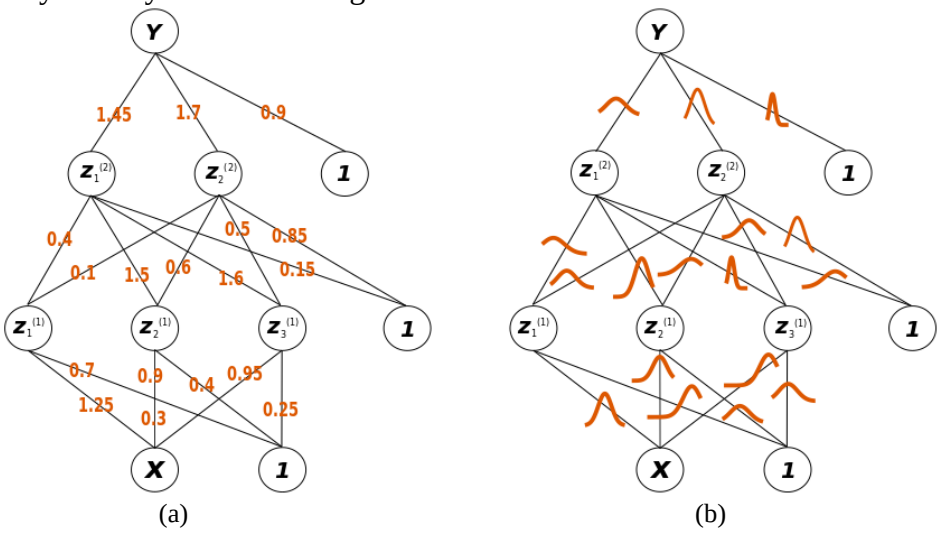

Figure 7: (a) Multi-Layer Perceptron with deterministic weights versus (b) Bayesian neural network with stochastic weights

At inference time, prior distributions of parameters (namely $P(W)$) are transformed into posterior distributions performing multiplication by the likelihood and division by a suitable normalizing constant to guarantee proper distribution shaping. Under such operations, the background knowledge before observations (e.g. preference for simple models to avoid overfitting) is converted into specific information about parameters acquired through available samples.

In mathematical terms, the posterior distribution of the weights for the EPF neural network is given by the following expression:

$$P\left(W\middle|\left(X^{(1)},y_{PUN}^{(1)}\right),\dots,\left(X^{(N)},y_{PUN}^{(N)}\right)\right)=\frac{L\left(W\middle|\left(X^{(1)},y_{PUN}^{(1)}\right),\dots,\left(X^{(N)},y_{PUN}^{(N)}\right)\right)P(W)}{P\left(\left(X^{(1)},y_{PUN}^{(1)}\right),\dots,\left(X^{(N)},y_{PUN}^{(N)}\right)\right)} \quad (1)$$

In the expression above, $X^{(i)}\in R^{d}, y_{PUN}^{(i)}\in R,\ i=1,\dots,N$ represent respectively the input features set of size d and day-ahead prices values (e.g. forecast targets) over a data set of size N, whereas W represent the model parameters. The specific shape of the parameters vector depends on the configuration of the neural network in use.

In practice, the parameters distributions are reshaped within the posterior. In fact, most likely values within the weights distributions are amplified during training in spite of less probable regions. Consequently, posterior distributions become more concentrated closer to values related to EPF/conditioning variables observations.

The combination of prior beliefs and likelihood within the learning procedure represents the cornerstone to introduce weights-related prediction uncertainty within the EPF-neural network in spite of the usual maximum likelihood point estimate. Indeed, learnt posterior distribution can be employed to achieve a prediction distribution over the provided data set.

The major aim of EPF is to learn the best predictive distribution for out-of-sample forecast given related inputs. To this end, the posterior fitted onto the training data must be implemented within a Bayesian inference machinery, by integrating the EPF neural network predictions with reference to the posterior:

$$P\left(y_{PUN}^{(N+1)}\middle|X^{(N+1)},\left(X^{(1)},y_{PUN}^{(1)}\right),\dots,\left(X^{(N)},y_{PUN}^{(N)}\right)\right)=\int P\left(y_{PUN}^{(N+1)}\middle|X^{(N+1)},W\right)P\left(W\middle|\left(X^{(1)},y_{PUN}^{(1)}\right),\dots,\left(X^{(N)},y_{PUN}^{(N)}\right)\right)dW \quad (2)$$

The aim of the integral is to marginalize out all the uncertainty embedded within EPF model parameters. By formulation above, EPF is achieved through a weighted sum of neural networks models. Each model is characterized by specific values for the weights. Therefore, most probable models (i.e. with higher evidence) are mostly considered within the prediction. Indeed, an interesting by product of Bayesian inference is the introduction of a quite natural form of regularization by integrating out the parameters instead of optimizing them. Furthermore, compared to maximum penalized likelihood based methods to reduce overfitting, it supports in-training phase regularization (i.e. no formal need to split training and validation sets to configure regularizers hyperparameters), faster cross-validation over large hyperparameters sets while also providing prediction uncertainty indications [76].

Despite reported advantages, the straight integration over the posterior results computationally intractable for the problem at hand. Therefore, practical implementations of probabilistic EPF required the application of a proper solution approximation, as detailed in Section 3.5.

3.2. Likelihood specification and observation noise treatment

As regarding the likelihood specification, in this work we followed the common Gaussian noise hypothesis over prediction residuals (see e.g. [67],[3] for EPF) stating $y^{(i)}=f\left(X^{(i)}\right)+\varepsilon^{(i)}$, with $\varepsilon^{(i)}$ characterizing independent random error samples from a Gaussian noise distribution. Alternative noise distributions will be considered within future extensions of the present work. Therefore, the conditional distribution for the predictions (i.e. price values) given the input features (i.e. past values of the price and other conditioning variables) is defined as a factorized Gaussian with mean $f_{DNN}(X,W)$ (i.e. neural network output), leading to the following regression likelihood for EPF forecast:

$$P(y_{PUN}|\,X,W)=\frac{1}{\sqrt{2\pi}\sigma}\,e^{-\frac{1}{2\sigma^{2}}\left(f_{DNN}(X,W)-y_{PUN}\right)^{2}} \quad (3)$$

The distributions of conditioning variables are not modeled themselves, as common for supervised learning contexts as EPF. Then, following the standard independence assumption (see e.g. [3]) between EPF errors $\varepsilon^{(i)}$ over different samples, by exploiting a conditioning variables set including

values of the price for previous days as described in following sections, the conditional likelihood related to model forecasts over training data factorizes as follows:

$$L\left(W\middle|\left(X^{(1)},y_{PUN}^{(1)}\right),\dots,\left(X^{(N)},y_{PUN}^{(N)}\right)\right)=\prod\prod_{i=1}^{N}P\left(y_{PUN}^{(i)}\mid X^{(i)},W\right) \tag{4}$$

Usually, the variance of the residual error is considered fixed or eventually as a configurable hyper parameter within maximum likelihood based neural network regression. Indeed, recently proposed methods for EPF deploying neural networks follow such assumption (see e.g. [20], [12],[10]). Nevertheless, fixed residual variance assumption is typically effective only when processing time series with homoscedastic behavior for every input point.

Therefore, considering the typical heteroscedastic characteristics of day-ahead price, proper preprocessing must be performed to cast an almost homoscedastic time series. In this study, we decided to follow a different approach, implementing a forecasting system originally supporting heteroscedasticity. The developed model implicitly assume observation-dependent noise characterized by specific variance levels. To this end, we deployed a Gaussian regression model including a specific deep neural network $g_{DNN}(X^{(i)},W')$ meant to learn the noise variance function over the observed data. Furthermore, we parametrized the standard deviation pointwise by a soft-plus operator:

$$\sigma^{(i)}=\log\left(1+\exp\left(g_{DNN}(X^{(i)},W')\right)\right) \tag{5}$$

In this way, we constrained the learning process to fit a proper variance function (i.e. is always non-negative).

It is worth remarking in this section how Bayesian learning for EPF relates to common maximum likelihood training methods. In fact, taking the log of the likelihood and assuming heteroscedasticity, we get the following equation including the Euclidean loss over the parameters and a term related to the changing variance:

$$Loss_{ML}=\sum_{i=1}^{N}\left[\frac{1}{2}\ln\left(2\pi\sigma^{(i)}\right)+\frac{1}{2\sigma^{(i)}}\left(f_{DNN}(X^{(i)},W)-y_{PUN}^{(i)}\right)^{2}\right] \tag{6}$$

Then, performing the common homoscedasticity assumption (i.e. constant variance), we obtain the sum of squared error loss usually adopted within maximum likelihood based training methods for EPF problems:

$$Loss_{ML}=\sum_{i=1}^{N}\left(f_{DNN}(X^{(i)},W)-y_{PUN}^{(i)}\right)^{2} \tag{7}$$

We may call attention here to a strength of Bayesian learning for EPF. Maximum likelihood training by Euclidean loss typically provides as output a parameter vector estimation representing a local minimizer of a complex function with several local minimum and saddle points. When a huge amount of training data is available, such minimizer usually provides a good estimate of the mapping function parameters. On the other hand, when limited samples are accessible, as in EPF, the adoption of a method providing a distribution of network weights could help users understand what the network has learnt from the available observations. Moreover, visualization and detailed analysis of the shapes of the learnt distributions (i.e. weights variance, etc.) could provide useful insights to validate the procedure and investigate the major characteristics of the model before exploitation for out-of-sample predictions. Furthermore, active learning processes can be deployed, analysing the evolution of parameters distribution by integrating new observation during time.

3.3. EPF point estimate from regression distribution

As introduced within the previous section, integration of likelihood times the posterior distribution of the deep neural networks parameters provides a prediction distribution for EPF. Nevertheless, in some circumstances, practitioners would require single value predictions. As an example, modern energy-aware planning and scheduling systems exploit predicted energy price profiles to calculate

the optimal strategy (see e.g. [11]).

Deep learning-based EPF with maximum likelihood training provides a single point estimate, thus the implementation comes naturally. Under a Bayesian framework, the proper way to extract point estimates could be less evident. In fact, the correct choice strictly depends on the loss function (e.g. prediction error) chosen, expressing a specific judgment to be applied to the forecast deviation compared to the target value [77]. As an example, the mean of the predictive distribution minimizes the expected value of the regression cost function when sum of squared error loss is preferred. On the other hand, the predictive distribution's median represents the best choice when facing absolute error related criteria.

In our case, the following point-estimate formulation is obtained:

$$\hat{y}_{PUN}^{(N+1)} = \int f_{DNN}\left(X^{(N+1)}, W\right) P\left(W \middle| \left(X^{(1)}, y_{PUN}^{(1)}\right), \dots, \left(X^{(N)}, y_{PUN}^{(N)}\right)\right) dW \tag{8}$$

In general terms, a Monte Carlo approximation over the posterior distribution might be performed, by generating a proper number of samples. Then, beyond the single-value energy price guess, the posterior samples can be analysed to evaluate related characteristics (e.g. dispersion).

Practically, Bayesian prediction in out-of-sample conditions is often performed by means of an expectation over the posterior distribution. It is worth noting that performing EPF by full expectation can be considered as implementing an ensemble of an infinite number of deterministic neural networks, governed by the posterior distribution [76]. Under maximum likelihood training approaches, such expectation is approximated by a single (i.e. most likely) value of the parameters. In Bayesian learning by Monte Carlo approximation of the expectation, EPF point estimates can be calculated as follows, where each W_t with $t = 1, \dots, T$ represents a sample of weights from the posterior distribution and D represent both input and target data for simplicity:

$$\hat{y}_{PUN}^{(N+1)} \approx {}^{1}/_{T} \sum_{t=1}^{T} f_{DNN}\left(X^{(N+1)}, W_t\right), \quad W_t \sim P(W|D) \tag{9}$$

3.4. EPF forecast by posterior approximation and inference

Despite the reported potentialities of Bayesian deep learning for EPF, finding the distribution over the weights parametrizing the network given the observations could be quite challenging to be achieved. Indeed, the posterior distribution of the weights is typically very complex for articulated models as neural networks. Therefore, exact Bayesian inference on the weights would be computationally intractable for EPF applications with reasonable network size. Indeed, in recent years an extensive research effort has been devoted to tractable approximate inference and several methods have been proposed within the recent literature on Bayesian learning. Developed techniques represent special cases from two major classes: sampling-based and variational inference based approaches.

Within the first class, most techniques exploit extension of the well-known Markov Chain Monte Carlo (i.e. MCMC) method. Notably, sampling-based techniques do not require the statement of assumptions about the posterior form, perhaps representing their major strength. In particular, compared to factorized variational inference techniques discussed later, sampling provides theoretical capability to learn correlations between different parameters. Nevertheless, MCMC-based approaches usually turn out critically slow in learning. To speed up convergence, authors of [77] deployed a systematic traversal of posterior space by Hamiltonian dynamics (i.e. HMC). Nonetheless, exploitation of HMC could be very challenging in practice, mainly due to intricate configuration of leapfrog step-size and burn-in period length. To overcome such limitations and improve scalability to large data sets, [78] proposed Stochastic Gradient Langevin Dynamics approach, adopting a single leapfrog step and a stochastic estimate of the likelihood gradient. Nevertheless, proposed methodology still manifest several weaknesses in practice. Indeed, quite correlated samples are often generated, thus requiring the integration of proper data discarding techniques. Most notably, learning usually tends to converge to single modes of the posterior distribution. Such an effect is often caused by the sensibly fast decrease of the step size needed for averaging out gradient stochasticity and cutting-out Metropolis-Hastings rejection rate. Hence, the major theoretical strength of such methods,

compared to factorized variational inference, usually turns out un- exploitable in practice.

Considering the open issues reported above, we focused on variational inference-based techniques within the proposed probabilistic EPF framework. The variational distribution is conceived to provide a proper approximation to the posterior while being easier to evaluate. To this end, most methods attempt to minimize the Kullback–Leibler divergence (i.e. relative entropy) between the two distributions, usually reframed in the following form:

$$D_{KL}(q(W,\lambda)||P(W|D)) = -\int q(W,\lambda)\, log\left(\frac{P(W|D)}{q(W,\lambda)}\right) dW = -\int q(W,\lambda)\, log\left(\frac{P(W,D)}{q(W,\lambda)}\right) dW + logP(D) \quad (10)$$

In the above equation, $P(W|D)$ represents the posterior distribution of EPF network weights, whereas $q(W,\lambda)$ a variational distribution parametrized by λ. Hence, since the evidence of EPF data P(D) is unknown and intractable but constant with respect to the variational parameters λ, it results that the first term on the right-hand side of the equation controls the Kullback–Leibler divergence. Such term is usually referred to as negative variational free energy or Evidence Lower Bound (ELBO).

To achieve practical solutions to our EPF problem, ELBO expression must be rewritten in the following form, which is obtained applying Bayes rules and Monte Carlo estimation to the integral:

$$ELBO(\lambda) = \int q(W,\lambda) log\left(\frac{P(W,D)}{q(W,\lambda)}\right) dW$$

$$= \int q(W,\lambda) log\left(\frac{P(W)}{q(W,\lambda)}\right) dw + \int q(W,\lambda) \log P(D|W) dW$$

$$= -D_{KL}(q(W,\lambda)||P(W)) + E_{q(W,\lambda)}[\log P(D|W)] \quad (11)$$

The first term, namely the KL divergence between the prior and variational posterior distributions of the weights, acts as a penalizer over complex posterior distributions by forcing $q(W,\lambda)$ to be closer to the prior. The second term, namely the expectation of the log-likelihood distribution, drives the optimizer to fit values explaining the available observations. Hence, an Occam razor mechanism is naturally integrated within ELBO maximization.

The following step to set up the Bayesian EPF framework was the specification of the class of variational distributions to be implemented within the training algorithm. It is worth remarking that, depending on the characteristics of the distributions provided to the learning process, closer or looser fits might be achieved with different computational cost. Several approaches have been proposed within the research literature, each with specific strengths and weaknesses [79]. In this work we followed the Minimum Description Length based technique proposed by [80], relying on a fully factorized Gaussian posterior to simplify computations:

$$q(W,\lambda) = \prod_{j=1}^{N_W} q(W_j,\lambda) = \prod_{j=1}^{N_W} N(W_j\,;\mu_{Wj}\,,\,\sigma_{Wj}^2\,) \quad (12)$$

Nevertheless, straight computations of the gradient of expected log-likelihood within ELBO optimization resulted too computationally intensive for realistic deep neural network dimensions. On the other hand, direct application of common Monte Carlo estimation to the ELBO gradient manifested high variance, thus resulting ineffective for practical problems [81]. To address such issues, several techniques have been proposed (see e.g., [74], [75],[82]).

In the present work, we adopted the formulations proposed by [74], supporting an unbiased estimation of the gradient while leveraging on training data sub-sampling over mini-batches. To this end, the path-wise derivative estimator (i.e. re-parametrization trick) has been exploited. Explorations of alternative formulations are foreseen within future developments. The algorithm has been developed leveraging on previous results from [75] demonstrating that sampling can be reframed to Gaussian weights perturbation when the variational posterior is defined as a factorized Gaussian distribution. Therefore, auxiliary random variables have been integrated to network weights, characterized by independent marginal distributions generated by Gaussians in standard form (i.e. zero mean and unitary scale). Following transformation of the weights is then obtained:

$$w = \mu_w + \sigma_w \varepsilon, \varepsilon \sim N(0, I) \quad (13)$$

Subsequently, the intractable expectation of the log-likelihood distribution was converted into the following form:

$$ELBO(\lambda) \approx -D_{KL}(q(W,\lambda)||P(W)) + \frac{1}{S}\sum_{i=1}^{S} \log P\left(D|W^{(i)}\right), \;\; W^{(i)} = \mu_{w^{(i)}} + \sigma_{w^{(i)}}\varepsilon, \;\; \varepsilon \sim N(0, I) \quad (14)$$

Practically, the latter term of the ELBO expression resulted computable by applying random perturbations to the usual log-likelihood loss adopted for frequentist style training, encapsulated within a Monte Carlo estimation loop. It is worth noting that the implemented weights perturbations by Bayesian learning provides an effective procedure to explore the most relevant parts of the parameter space, averaged out by EPF.

Afterwards, considering the results of [75], the objective function can be reframed into a mini-batch training process for mini-batch $m = 1,\ldots,M$ as:

$$OBJ_{mb} = \left[\frac{1}{M} D_{KL}(q(W,\lambda)||P(W)) - \frac{1}{S}\sum_{i=1}^{S} \log P\left(D^{(m)}|W^{(i)}\right)\right] \quad (15)$$

Here, M represents the size of the sub-samples from the training data used to estimate the objective function gradient. Basically, mini-batch training performs an estimation of the expected value of the objective function by randomly subsampling and averaging small subsets of the training examples. In fact, computing the expectation on the overall training set often becomes too computationally expensive and constrained by available memory. Moreover, several mini-batches can be processed in parallel, e.g. by leveraging on the availability of multi-core Graphics Processing Units. In this way, overall training time is strongly reduced.

It is worth nothing that the Bayesian deep learning formulation deployed within the proposed probabilistic EPF framework naturally provides a facility of practical significance. Indeed, training can be performed within the widely available development environment for deep learning (i.e. Tensorflow, PyTorch, etc.) using the same solvers usually adopted for maximum likelihood based training. In particular, we exploited the Adam algorithm [83], implementing adaptive estimates of lower-order moments to deal with non-stationary objectives and very noisy and/or sparse gradients. Further details (e.g. solver set-up, configuration, etc.) will be provided within the results section.

3.5. Deployed deep neural network architecture and integration within the Bayesian framework

The next component to be defined within the Bayesian EPF framework is the architecture of the neural network. Indeed, the Bayesian deep learning approach reported above is agnostic to the specific form of the model. As introduced within Section 1, several architectures have been considered for EPF within previous research studies, including feedforward and recurrent networks. For this work, we exploited the Deep Neural Network architecture recently proposed by [65]. Indeed, the paper reports a detailed empirical investigation of alternative architectures including recurrent neural networks and convolutional neural networks. Then, authors showed that the DNN (i.e. an extension of the Multi-Layer Perceptron) provided better results over Belgian day ahead market. Detailed investigations of Bayesian reformulation of alternative architectures (e.g. Bayesian LSTM) are foreseen as future study. In general, a deep network is expected to be more effective than a shallow alternative due to the theoretical capability to learn hierarchical mappings and layer-wise multi-scale representations. Therefore, the DNN is envisioned to identify patterns on different lags depending on the features. Perhaps, such capability of deep networks could strongly support day-ahead forecast. Indeed, the dynamics of day-ahead market prices are usually characterized by pattern of intra-day nonlinear functions of fundamental variables (see e.g., [84]).

The subsequent step to be performed is the specification of the Bayesian layers within the neural network. Several alternatives have been investigated within the research literature (see. e.g., [85]). For the present study we deployed the network form employed in [86] and [101], based on the hierarchical composition of deterministic and stochastic layers. The rationale behind such decision is twofold. On the one hand, we reduced the model complexity with related computational effort. On

the other hand, the deterministic layers are meant to extract features to be then exploited by the stochastic layer, mimic manifold Gaussian processes [87]. Alternative network configurations are left to future extensions.

The major architectural elements of the implemented deep neural network are briefly summarized hereafter, leaving further information to available literature (see e.g. [88]). In details, the deep neural network is composed by a hierarchical aggregation of layers. Latent features extracted by the hidden layers are then exploited by upper layers, thus supporting characterization of complex output mappings. Sigmoid and hyperbolic tangent activation functions represent the most exploited historically. Rectified linear units (analogous to Half-wave rectifiers) are often considered nowadays due to the simplified calculations during training. On the other hand, linear activation functions are usually employed within output layers for regression problems. In this study, we implemented Leaky ReLU activations instead of straight ReLU to enable positive gradients when the neuron is not active, thus mitigating potential Dying ReLU issues. Indeed, ReLU neurons could be trapped into perpetually inactive states inhibiting gradients flow during training.

The mathematical formulation of the deep neural network is reported hereafter. An architecture with two hidden layers and a single-value linear output layer is displayed for simplicity, whereas:

- $X_j^{(i)}$: j-th feature within i-th input data sample
- n_{in}: size of features vector provided as input to the network
- n_{h1}, n_{h2}: number of neurons within hidden layer 1 and 2
- $W_{j,h}^{(1)}, W_{h,k}^{(2)}, W_{k,1}^{(3)}$: network weights in each layer
- $W_{0,h}^{(1)}, W_{0,k}^{(2)}, W_{0,1}^{(3)}$: network biases in each layer
- W: vector aggregating the parameters of the network layer-wise
- $z_h^{(1)}, z_k^{(2)}$: outputs of the neurons in each hidden layer
- $f_h^{(1)}, f_k^{(2)}$: hidden layers activation functions

$$f_h^{(l)}(a_h) = \begin{cases} a_h & if\ \ a_h > 0 \\ 0.01a_h & otherwise \end{cases} \tag{16}$$

$$z_h^{(1)} = f_h^{(1)}\left(\sum_{j=1}^{n_{in}} W_{j,h}^{(1)}\, X_j^{(i)} + W_{0,h}^{(1)}\right) \tag{17}$$

$$z_k^{(2)} = f_k^{(2)}\left(\sum_{h=1}^{n_{h1}} W_{h,k}^{(2)}\, z_h^{(1)} + W_{0,k}^{(2)}\right) \tag{18}$$

$$f_{DNN}(X^{(i)}, W) = \sum_{k=1}^{n_{h2}} W_{k,1}^{(3)}\, z_k^{(2)} + W_{0,1}^{(3)} \tag{19}$$

3.6. Neural network inputs setup

Unquestionably, prediction accuracy is strongly influenced by the integration of conditioning variable to be exploited by the model. Indeed, several exogenous variables have been considered within the state of the art. For example, [44] included load and wind power generation while tackling residual autocorrelation and seasonal dynamics, showing usefulness to forecast on Western Danish price area of Nord Pool. Authors of [89] integrated a probabilistic representation of price peaks, demonstrated on German EEX market. Authors of [90] identified the need to tackle long-term information within the model for Nord Pool market, claiming relation to significant amount of supply from hydropower plants. [91] considered day-ahead load forecast and three dummies to represent weekly seasonality, with application to California power market. [92] exploited publically available market information on hourly Ontario energy price (HOEP). [93] adopted as reservoir levels and load in Colombian market. [25] included Nordic demand and Danish wind power generation whereas [30] exploited also air temperatures series.

Besides past price values, representing commonly adopted inputs to EPF models, the most effective

setup of further exogenous variables seems strongly related to the specific market conditions. Due to the lack of previous studies on Italian PUN price, we performed a first analysis by evaluating mutual Pearson Correlation Coefficients (PCC) between further exogenous variables available and the price time series. Actually, PCC is meant to identify linear correlations between the variables. Therefore, we investigated also the Maximal Information Coefficients (i.e. MIC) to discover eventual further relationships between variables. For such purpose, we adopted TICe and MICe estimators recently proposed by [94] and made freely available within minepy [95]. In particular TICe (total information coefficient) supports the assessment of significant relations between the variables whereas MICs support strength-based classification.

For the present study, we adopted the same approach proposed in [65] by focusing only data freely available on the European markets platforms. The exploration of further conditioning variable (e.g. temperature and wind speed forecast, gas price, etc.) is foreseen within future developments of the present study. Besides, weather and seasonal related phenomena might be partially considered by means of load forecast, as reported by [10]. In particular, we explored predictions available ex-ante within the Italian market websites. Specifically, predicted day ahead load (i.e. Load), overall generation (i.e. Gen), solar and wind power production (i.e.Solar, Wind) on the different sub-regions (i.e. CNOR, CSUD, NORD, SARD, SICI, SUC) as well as at national level (i.e. ITA). Further details on the available predictions can be found within Italian market and Terna websites (see e.g. [71]).

Table 2 reports the results obtained for the Italian market. As regarding Belgian market, we adopted the same set of predictors exploited within [65] in order to simplify comparability of obtained results. PCC and MIC are reported for consistency in Table 3.

Table 2. PCC, TICe and MICe on Italian PUN

Pearson Correlation Coefficient																
	ITA Load	ITA Gen	ITA Solar	ITA Wind	CNOR Solar	CNOR Wind	CSUD Solar	CSUD Wind	NORD Solar	NORD Wind	SARD Solar	SARD Wind	SICI Solar	SICI Wind	SUD Solar	SUD Wind
ITA Pun	0.59	0.68	-0.06	-0.13	-0.05	-0.04	-0.04	-0.13	-0.06	-0.12	-0.09	-0.10	-0.06	-0.08	-0.05	-0.11
TICe																
	ITA Load	ITA Gen	ITA Solar	ITA Wind	CNOR Solar	CNOR Wind	CSUD Solar	CSUD Wind	NORD Solar	NORD Wind	SARD Solar	SARD Wind	SICI Solar	SICI Wind	SUD Solar	SUD Wind
ITA Pun	4.02	5.50	0.32	0.28	0.27	0.14	0.28	0.30	0.27	0.33	0.43	0.22	0.32	0.18	0.22	0.24
MICe																
	ITA Load	ITA Gen	ITA Solar	ITA Wind	CNOR Solar	CNOR Wind	CSUD Solar	CSUD Wind	NORD Solar	NORD Wind	SARD Solar	SARD Wind	SICI Solar	SICI Wind	SUD Solar	SUD Wind
ITA Pun	0.23	0.30	0.03	0.03	0.02	0.02	0.03	0.03	0.02	0.03	0.04	0.02	0.03	0.02	0.02	0.02

Table 3. PCC, TICe and MICe on Belgian day-ahead market

Pearson Correlation Coefficient					
	BEL Load	BEL Gen	FRA Past Price	FRA Load	FRA Gen
BEL Price	0.52	0.18	0.74	0.39	0.32
TICe					
	BEL Load	BEL Gen	FRA Past Price	FRA Load	FRA Gen
BEL Price	3.32	0.41	7.06	1.74	1.64
MICe					
	BEL Load	**BEL Gen**	**FRA Past Price**	**FRA Load**	**FRA Gen**
BEL Price	0.18	0.03	0.36	0.11	0.11

4. Results

In this section we report the results obtained by the application of the proposed framework for probabilistic EPF to Italian and Belgian day-ahead markets.

To this end, we exploited Tensorflow open source machine learning environment [96]. Moreover,

we included the recently developed Tensorflow Probability library [97], providing several facilities including trainable probabilistic layers, prior distributions, Kullback–Leibler divergence, etc. As introduced within Section 3, we employed Adam algorithm to train the networks. Training epochs were set to 500. Moreover, we included a patience callback executed after 50 epochs. In this way, training procedure is interrupted in case the learning curves stop decreasing, without waiting until the last epoch. Minibatch size has been configured to 32 data samples, representing a good compromise between proper gradient estimation and computational burden.

Afterwards, overall datasets have been split into training, validation and test subsets. As introduced within Section 2, we left an entire year of samples to both testing and validation in order to perform experiments over all month specific conditions. We might remark that we did not perform data shuffling (i.e. random data reordering to support random distributions) before subset separation, as common for time series forecasting problems. In this way, we avoided the unfair integration of samples coming from time periods ahead, which often results in over-accurate predictions during validation compared to out-of-sample forecast.

Afterwards, we implemented a time-series cross validation procedure (i.e. rolling windows) instead of the conventional cross validation [98].

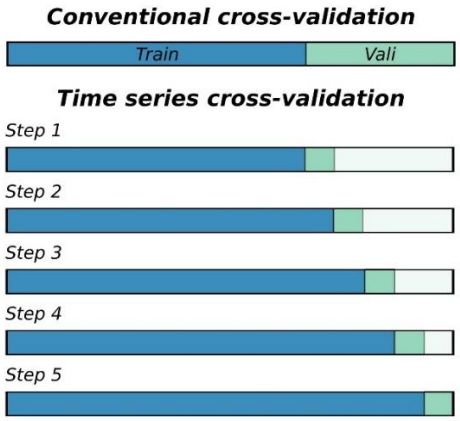

Figure 8: Time series cross validation procedure

By the conventional approach, validation is performed block-wise following training, leveraging on fixed subsets. On the other hand, time series cross validation exploits a sub split of the validation set into a configurable number of folds. The algorithm then proceeds as sketched in Figure 8. Network parameters are learned over the predefined training set and validated on the first fold. Afterwards, samples from the fold used for validation are integrated within the training set. Network training is re-executed and the next fold is employed as validation. The procedure continue until the last fold of the validation set is processed.

Perhaps, the major strength of time series cross validation is the extension of training set size, useful on problems as EPF characterized by a limited amount of observations. Moreover, the network is validated on data coming from time periods closer to the available set, thus partially compensating eventual short-term deviations within the generating process. Furthermore, practical model employment is emulated. Indeed, once the EPF system is put in production (i.e. used within an organization to perform out-of-sample prediction) it is quite common in practice to retrain it in order to consider last observations from the market, thus updating the model accordingly.

Afterwards, we performed datasets standardization by removing mean and scaling samples to unitary variance. Indeed, such operation provides a properly conditioned form to the set, fundamental to achieve an effective training of the Bayesian neural network. Considering the hour specific distributions, the operation has been performed separately.

To evaluate the accuracy of EPF model predictions, several indicators might be considered. As observed by [3], no commonly adopted "industry standard" exist for EPF, which may impact evaluations consistency and comparability between different studies.

Therefore, we compared the results using different indicators commonly adopted within spot

market field: Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) and Symmetric Mean Absolute Percentage Error (SMAPE).

RMSE expresses the squared root of the second sample moment of the residuals, but tend to be sensitive to large errors and outliers. Mean Absolute Error provides a different view, influenced in proportion to the absolute value of prediction errors. Since absolute error indicators might be difficult to be compared over different studies (e.g., due to different rescaling), we included the MAPE so to provide a relative view on the residuals. However, MAPE can be distorted by small values, becoming very large regardless the actual value of residuals. Moreover, MAPE values turns out to be small when processing higher electricity prices, irrespective of the absolute errors [1]. Nevertheless, MAPE is often adopted within EPF studies. Therefore we included it within performed results analysis.

Moreover, to overcome the reported limitations of MAPE, we comprised also SMAPE calculations. Such indicator exploits lower and upper bounds and mitigates sensitivity of MAPE to values close to zero, for which related contribution become very large and dominate final value. Conventional SMAPE formula expresses results in a range 0-200% which could be misleading to be interpreted. An alternative formulation is sometimes used in practice, by applying a 0.5 scaling factor within the denominator, which might provide more intuitive indications. Nevertheless, in the present study we employed conventional SMAPE equation in order to provide results comparable with previous studies on Belgian market (i.e. [65]). Summarizing, we implemented following indicators calculations:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_{PUN}^{(i)} - y_{PUN}^{(i)}\right)^2} \tag{20}$$

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_{PUN}^{(i)} - y_{PUN}^{(i)}}{y_{PUN}^{(i)}}\right| \tag{21}$$

$$SMAPE = \frac{100}{N}\sum_{i=1}^{N}\frac{\left|\hat{y}_{PUN}^{(i)} - y_{PUN}^{(i)}\right|}{\left(\left|\hat{y}_{PUN}^{(i)}\right| + \left|y_{PUN}^{(i)}\right|\right)/2} \tag{22}$$

Besides, following the recommendations reported in [66], we performed a quantitative evaluation of probabilistic forecasts by employing the Continuous Ranked Probability Score. CRPS summarizes calibration (i.e. forecast error) and sharpness (i.e. distribution concentration) within a unique measure (see, e.g., [102]):

$$CRPS\left(\widehat{F_P}, y_{PUN}\right) = \int_{-\infty}^{\infty}\left(\widehat{F_P}(z) - \mathbb{I}_{\{y_{PUN}\leq z\}}\right)^2 dz = \mathbb{E}_{\widehat{F_P}}\left|\hat{y}_{PUN} - y_{PUN}\right| - \frac{1}{2}\mathbb{E}_{\widehat{F_P}}\left|\hat{y}_{PUN} - \hat{y}'_{PUN}\right| \tag{23}$$

Given a test set of size N and M independent random samples from the probabilistic model, CRPS can be approximated by the following expression:

$$CRPS = \frac{1}{NM}\sum_{i=1}^{N}\sum_{s=1}^{M}\left|\hat{y}_{PUN}^{(i)(s)} - y_{PUN}^{(i)}\right| - \frac{1}{2NM}\sum_{i=1}^{N}\sum_{s=1}^{M}\left|\hat{y}_{PUN}^{(i)(s)} - \hat{y}_{PUN}^{(i)(s)'}\right| \tag{24}$$

where $\hat{y}_{PUN}^{(i)(s)}$ and $\hat{y}_{PUN}^{(i)(s)'}$ represents two independent samples generated from the EPF distribution.

After the identification of forecast evaluation indicators, we proceed with the characterization of the neural networks. For the implementation of multi-period forecast, two main classes of techniques can be considered. The first approach, representing the most popular one, is based on single stage prediction models (see, e.g., [99]). Afterwards, while shifting the forecast window towards the next future stages, unavailable measures are replaced by previous stage predictions. While being simpler and typically more effective for short-term predictions (e.g., one or two hours ahead), such recursive procedure could generate quickly growing error accumulation when longer-term forecast are needed. Moreover, the error increment over stages could be very critical for strongly non-linear problems. Alternatively, other studies adopted stage-specific modeling approaches, developing and training different models for each stage over the prediction horizon. Specifically for day-ahead energy price forecast, a model for each 24-th hour of the next day is developed. The aim of this approach, mainly inspired by extensive adoption within demand forecast research field, is to exploit specific hourly patterns of conditioned or conditioning variables that could be represented by a specialized model tuning. The second, less popular, approach exploits a straight multi-stage forecasting horizon,

typically constituted by models exposing outputs including 24 steps for day-ahead energy price prediction (see e.g. [100]). Considering available studies evaluations, the major drawback of this method seems to be related to the difficulties in achieving accurate forecast within the overall horizon by using traditional statistical and computational intelligence models (e.g. SARIMAX, shallow neural networks). In fact, identification of complex relations between conditioning and conditioned variables over the sequences was very challenging.

Considering reported open issues, we first performed an empirical comparison of deterministic network architectures with the same hidden layers (as in [65]) but different output layer, having single hour and straight 24 steps outputs respectively. We observed similar results over the validation set, as detailed in Table 4. Therefore, we chose a network architecture with specialized single output in order to reduce number of parameters. Nevertheless, we passed to each hour-specific network the available forecast of conditioning variables in order to exploit possible latent information related to following lags (e.g., forecast of demand at 11:00 might influence price at 10:00). Of course, only data available at time of prediction are provided (e.g. no conditioning related to future lags for predicting price at 24:00). Afterwards, while performing experiments on the test set described below, we checked the performed assumption. Results are included in Table 4. It is worth noting that obtained measures were slightly different from [65]. Perhaps, this was due to the different dataset used for networks training and validation.

Table 4. Comparison of networks with single hour and multi-hour forecast on Belgian market

	VALI	**TEST**
RMSE DNN_SingleOut	8.51	11.47
RMSE DNN_MultiOut	8.47	11.58
MAPE DNN_SingleOut	14.26	15.48
MAPE DNN_MultiOut	14.20	15.87
SMAPE DNN_SingleOut	13.73	16.18
SMAPE DNN_MultiOut	13.61	16.40

Following the indications reported above, both deterministic and Bayesian networks have been configured with the same number of layers and neurons in each layer. After a straight grid search, we implemented 250 and 150 neurons within each layer respectively. We might remark here that, for the present study, we did not perform an extensive hyper-parameter search (e.g. number of layers, number of neurons in each layer, etc.). The rationale behind such decision is twofold. On the one hand, the major aim of the present work was to investigate the capabilities of Bayesian in contrast to common frequentist style deep neural network for EPF, independently from (i.e. given) specific network structure. On the other hand, we performed predictions on the Belgian market considering a deterministic deep network architecture previously proposed within research literature. Therefore, we meant to be unbiased by adopting coherent configurations. Nevertheless, we envision extensive grid search over hyperparameters space or investigations of more advanced approaches for such purpose (as e.g. in [10]) to future developments.

As regarding the set-up of models inputs, Section 3.7 reported a brief analysis of exogenous variables exploitable for Italian and Belgian markets. To properly set-up the EPF system, specific lagged values on each input series must be identified, to be then included as input features. To this end, dedicated selections are usually needed for each input variable depending on related correlations over the sequence. Indeed, some variables could require a linear window (e.g. last 3 lags) while others specific lags (e.g., lag-7, lag-24, etc.). Therefore, nonlinear windows should be implemented to avoid increase of problem dimension. In order to achieve this particular aim, detailed time series evaluations are often performed, typically adopting tools such as Pearson correlation coefficients analysis with related heat-maps plots, Autocorrelation (i.e. ACF) and Partial Autocorrelation (i.e. PACF) functions plots, etc. (see e.g., [12],[20]). Actually, an extensive assessment between various input series across all time steps are usually rather time consuming to be performed in practice. Indeed, day-ahead price

forecast usually involve quite a few exploitable series over several lags, thus generating multiple possible combinations to be analyzed in order to identify patterns. Indeed, partial exploration is often performed in practice, aimed to identify major components to be included within the model. Nevertheless, prediction accuracy is often strictly related to the quality of chosen features.

In this work we decided to follow a different approach. In particular, we were interested to investigate the capability of deep neural networks to automatically learn useful features from raw input data, represented as time series of exogenous input variables as well as past values of the hourly price. Foremost, we guessed that the capability of deep networks to learn statistical invariants could support the identification of day-ahead price series patterns during time, thus isolating actual correlations from random noise. Specifically, recognizable relations are expected within various days of the same period (e.g. month) between day-ahead prices and past values of hourly price and exogenous variables as well as available predictions of potentially conditioning variables (e.g. load forecast, etc.). In fact, as further detailed within data analysis section, day-ahead price profiles manifest quite evident behavior depending on time period and conditioning variables (e.g. peaks during higher demand hour, working vs nonworking days, etc.).

Due to the lack of internal memory (i.e. stateless), DNNs must receive past values of the time series as inputs to be considered within the model. To this end, a window of predefined length must be applied to the overall time series in order to extract evenly spaced batches of data. The size of such window (representing the feature-lags search space) was tuned by analyzing the last sensible lags within the correlation plots. Considering results from Section 2, in this study we selected a window width of 24 lags.

Moreover, as reported in the analysis section, day-ahead price time series exhibit seasonal behavior. To tackle such issue, we followed a previously proposed approach (see e.g. [65]), passing dedicated identifiers (including day and year's week identifiers) as input to the model. Therefore, the network is configured to learn specific period nonlinear mappings to conditioning variables (e.g. load) while avoiding very long horizons, critical for both computational and dataset size issues.

Following the definition of the conditioning variable set, we converted the overall data framework into a supervised learning problem by applying a sliding window technique.

Obtained quantitative results on the test sets are reported in Table 5 whereas Figures 9 and 10 display the errors distributions on Italian and Belgian markets respectively. Notably, the Bayesian deep neural network achieved performance comparable with the deterministic neural network.

It is worth noting that we observed slightly different results for the DNN on the Belgian market compared to a previous study employing a similar deterministic architecture [65]. This could be due to the different characteristics of employed datasets (training/validation on 01/01/2010 - 31/11/2015 and test on 01/11/2015 to 31/11/2016 in [65]). Further investigations on such issue are left to future extensions since outside the scope of present study.

Table 5. Performance on Italian and Belgian test set

Test set results		
	Ita	**Bel**
RMSE BNN	9.0	11.3
RMSE DNN	9.0	11.5
MAPE BNN	11.3	15.4
MAPE DNN	11.5	15.5
SMAPE BNN	11.5	16.1
SMAPE DNN	11.7	16.2
CRPS BNN	7.46	9.29

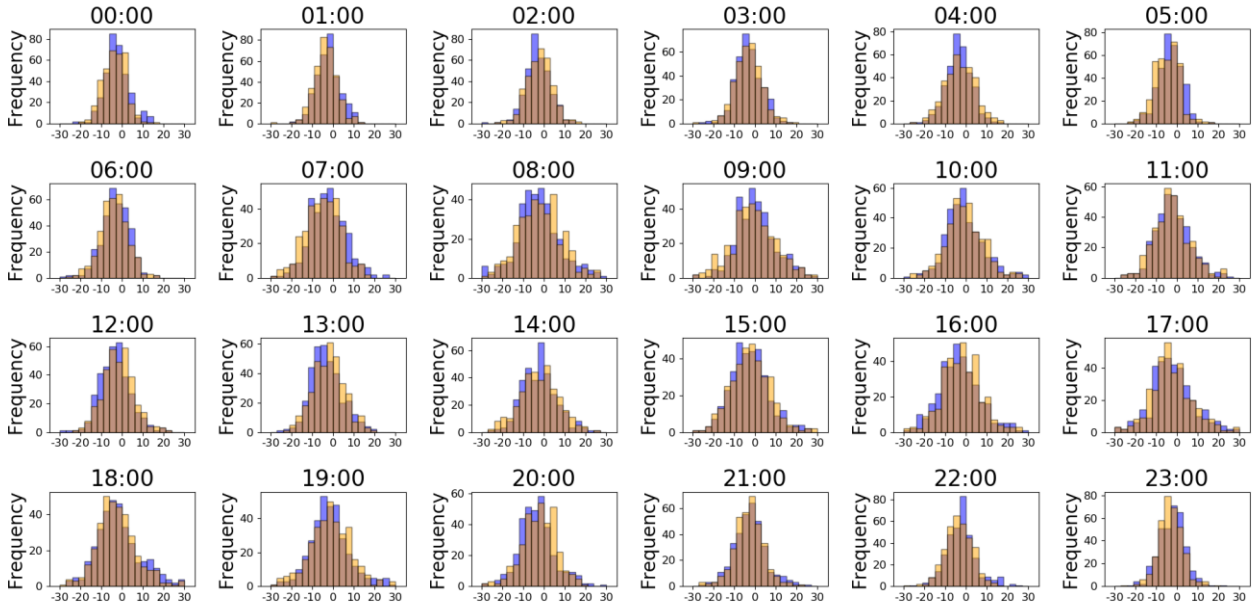

Figure 9: Distribution of errors on Italian market test set: DNN (orange), Bayesian DNN (blue)

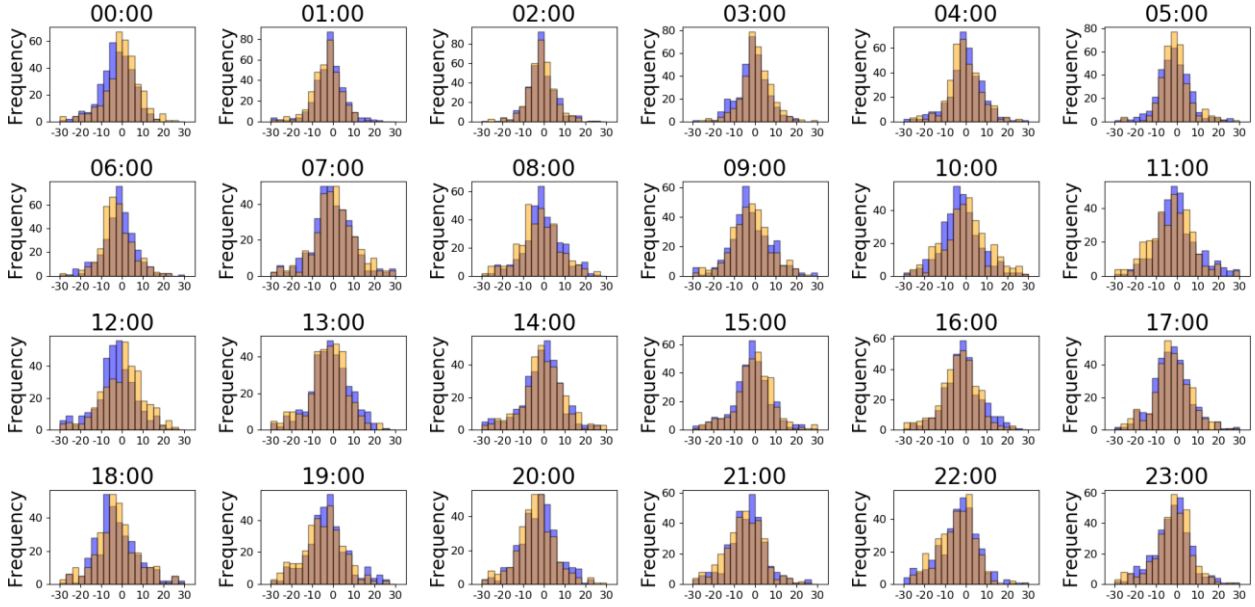

Figure 10. Distribution of errors on Belgian market test set: DNN (orange), Bayesian DNN (blue)

We might remark here that we do not consider the improvement in forecast accuracy as the major outcome of Bayesian deep learning for EPF. Actually, different executions of training algorithms usually result in small random fluctuations in performances while processing data sets characterized by limited samples as EPF. Such phenomena strictly depends on specific local minimum found by the solver within the quite broad set provided by complex network architectures.

In our opinion, the major strength of the proposed method resides in the provided distribution of predicted prices, extending the usual point forecast. Such information might be exploited in different ways depending on the specific application (see e.g., [66]). For example, by sampling from the distribution, alternative electricity consumption/production scenarios can be simulated within software tools for generation asset management (e.g. [113]) or energy-aware production scheduling (e.g. [11]), enabling planning and investigation of alternative strategies. Figure 11 displays examples of samples from the predictive distribution.

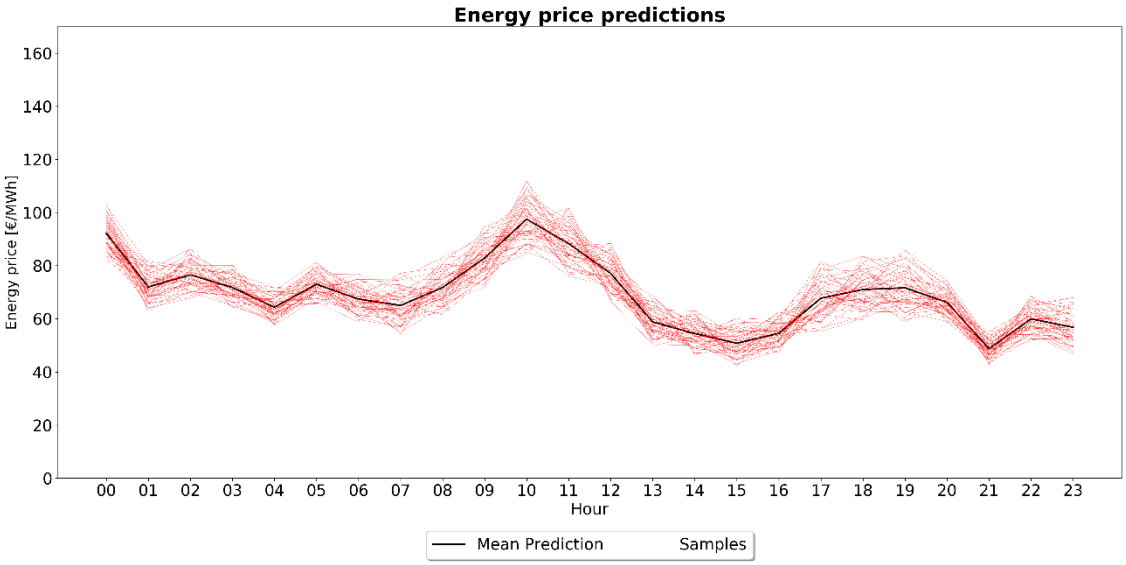

Figure 11: Examples of samples from the predicted distributions of day-ahead prices

Furthermore, hour-specific uncertainty indications represent fundamental ingredients to achieve advanced short-term risk management strategies (see, e.g.,[107]). Figure 12 illustrates such facility, provided by the distribution of predictions, over same test set samples related to different months. The displayed standard deviation were obtained straightforwardly by applying the posterior formulations reported in Section 3, including the heteroscedastic component.

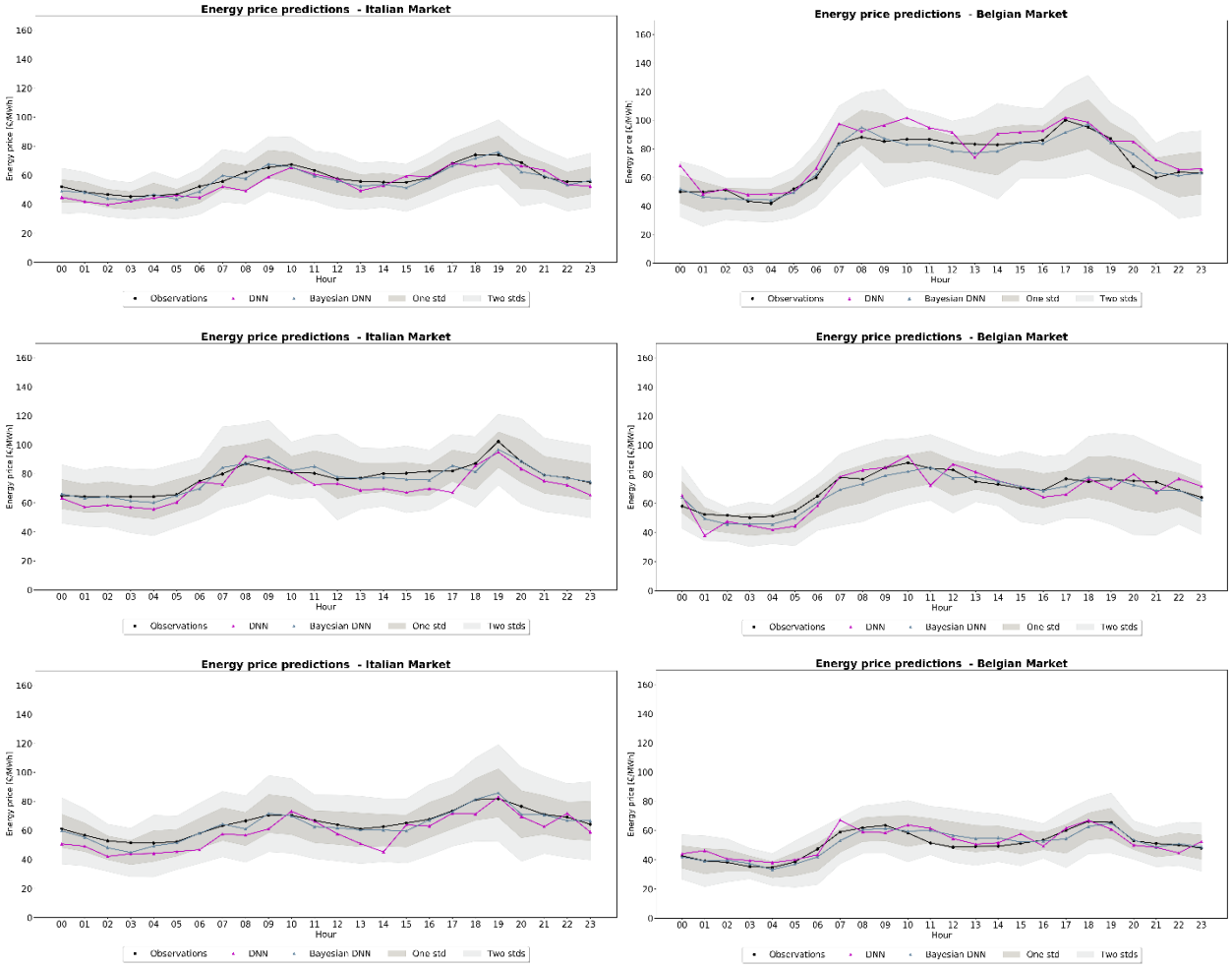

Figure 12: Predicted distributions over samples from the test set: Italian market (left), Belgian market (right)

In particular, we reported multiple examples for both Italian market (left part of the figure) and Belgian market (right part of the figure) displaying different operating conditions. For each subplot, we included also the predicted standard deviation on each hour. Notably, the Bayesian neural network provided predictions with different confidence levels depending on the specific day. Actually, EPF model outputs are strictly related to particular daily conditions, being closer or farther to the observations available within the training set or with different noise variances.

Hereafter, we report further quantitative insights into the results obtained on both markets. In particular, Tables 6-7 report performance indicators calculated on hourly aggregated samples. Tables 8-9 and Tables 10-11 (in appendix) report results achieved on different weekdays and months respectively.

Figures 13-15, displaying box and whiskers plots of Absolute Percentage Errors (i.e. APE), were included to provide insights into the distributions of errors within specific data aggregations (i.e. hourly, daily and monthly). It is worth noting here that APE is influenced by the scale of the error, thus being more pronounced with lower price values (e.g. 4:00 a.m.).

Notably, slight performance variations appear between different hours. As reported above, such effect could be caused by random convergences to specific local minimum during networks training. Detailed investigations might be performed to clarify the latent dynamics behind such observations, perhaps by in-depth analysis of residuals distribution over different networks hyperparameters or by application of dataset augmentation techniques. We left such extensions to future developments since outside the scope of present study.

Table 6. Italian market test set performances on different hours

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
RMSE BNN	6.4	6.6	6.6	7.1	7.2	6.6	7.4	9.0	12.1	11.1	9.3	9.2	8.4	8.3	8.6	10.1	11.3	10.5	11.9	11.5	10.3	8.9	7.9	6.7
RMSE DNN	6.6	6.6	6.6	7.2	7.9	7.8	8.0	10.0	11.1	11.8	8.9	9.3	8.0	8.0	9.4	10.0	10.6	10.9	10.8	11.3	9.0	8.1	7.7	6.9
MAPE BNN	9.0	10.0	8.6	11.0	10.4	12.2	9.7	12.9	9.1	11.6	9.5	13.0	13.0	11.6	11.5	11.3	13.5	10.5	12.0	12.3	10.7	13.7	12.1	13.0
MAPE DNN	9.1	10.1	10.9	13.1	13.8	11.8	11.0	12.3	12.5	13.3	11.2	11.6	10.9	11.9	14.1	13.9	13.1	12.1	11.3	11.1	9.4	8.9	9.4	9.0
SMAPE BNN	9.3	10.2	8.9	11.1	10.8	12.4	10.2	13.0	9.3	11.8	9.8	12.8	13.2	11.8	11.3	11.7	13.4	10.9	12.1	11.9	10.8	13.8	12.1	13.1
SMAPE DNN	9.5	10.4	10.9	12.8	13.8	12.5	11.5	13.1	12.9	13.2	11.2	12.0	10.9	12.0	13.9	13.8	12.9	12.3	11.5	11.3	9.7	9.3	9.8	9.4
CRPS BNN	4.6	4.7	4.8	5.3	5.2	4.8	5.3	6.6	8.3	7.5	6.5	6.5	6.0	6.2	6.2	7.2	8.0	7.4	8.2	7.5	7.2	6.1	5.5	4.7

Table 7. Belgian market test set performances on different hours

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
RMSE BNN	9.4	8.3	8.0	8.8	8.9	8.6	8.9	12.0	11.0	12.1	12.5	12.7	11.8	11.6	11.4	11.5	11.4	13.4	14.7	13.8	11.9	11.3	12.6	11.2
RMSE DNN	9.4	8.7	7.3	7.3	8.0	7.3	9.3	12.3	12.8	12.4	13.1	13.4	11.4	11.6	11.7	11.9	10.9	13.3	14.5	14.7	12.4	12.3	12.6	11.4
MAPE BNN	14.4	14.9	16.9	31.8	24.4	18.3	14.5	16.2	13.2	14.5	15.6	12.3	16.5	19.0	20.1	18.3	22.3	18.0	16.0	14.1	13.9	14.4	14.5	14.5
MAPE DNN	14.1	14.7	15.5	29.4	21.8	16.7	15.9	17.2	14.6	13.5	16.7	12.8	17.3	18.5	20.4	18.1	23.6	18.9	15.6	14.7	14.4	16.0	14.7	14.5
SMAPE BNN	15.3	14.9	15.7	19.4	20.4	17.8	15.1	16.2	13.5	14.7	15.8	16.0	15.7	16.8	17.3	18.0	17.3	16.2	16.0	14.5	14.2	15.1	15.3	14.9
SMAPE DNN	14.2	15.4	14.5	17.0	19.1	15.2	16.3	16.3	16.0	14.5	16.7	17.2	15.9	16.8	18.0	17.6	17.5	16.3	15.7	15.5	14.9	16.9	15.7	15.0
CRPS BNN	6.2	5.3	5.1	5.6	5.9	5.8	5.8	7.6	6.8	7.8	7.9	8.0	7.7	7.6	7.3	7.2	7.3	7.9	9.2	8.6	7.7	7.6	8.0	7.1

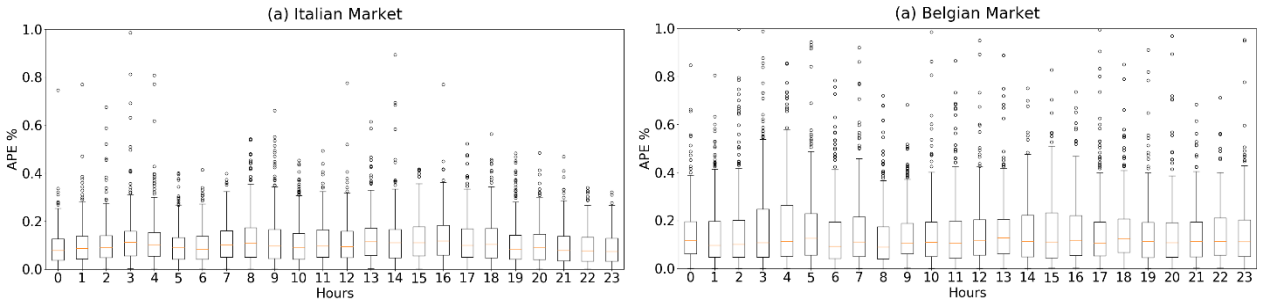

Figure 13: Hourly Absolute Percentage Error over test set: (a) Italian market, (b) Belgian market

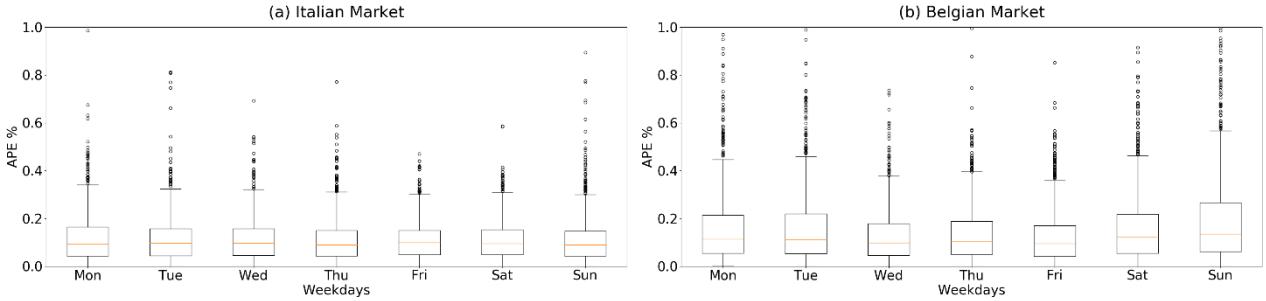

Figure 14: Daily Absolute Percentage Error over test set: (a) Italian market, (b) Belgian market

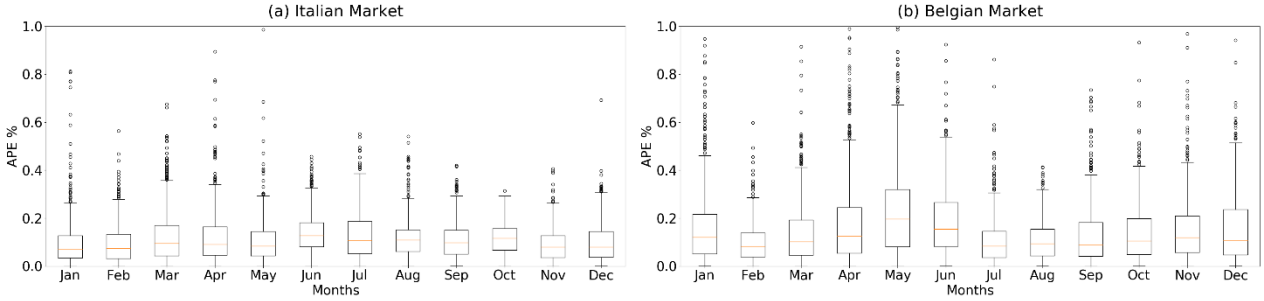

Figure 15: Monthly Absolute Percentage Error over test set: (a) Italian market, (b) Belgian market

5. Conclusions

In this paper we have presented a novel method to achieve probabilistic day-ahead electricity prices forecasting based on Bayesian deep learning. To this end, we deployed a Bayesian inference framework introducing probability distributions over neural network weights and a Gaussian likelihood function. Hence, we integrated the deep network forecasts with reference to the posteriors to achieve predictions in out-of-sample conditions. Such integration marginalized out the uncertainty embedded within model parameters and introduced a natural form of regularization. Actually, performing EPF by Bayesian deep learning provides an approximation to an ensemble of an infinite number of deterministic neural networks, governed by the posterior distribution. Moreover, the method enabled sampling from the predictive distribution and visualization of predictions uncertainties, supporting detailed forecasts analysis.

Besides, we extended the Bayesian regression model by including a specific deep neural network meant to learn the noise variance function, parametrized by a soft-plus operator to guide the learning process. Therefore, we obtained a forecasting system originally supporting heteroscedasticity, giving specific noise variances depending on the period.

Afterwards, we tackled the computational intractability of exact Bayesian inference on the deep network weights for EPF applications with reasonable network dimensions. To this end, we employed parametrized variational distributions within a variational inference framework. Then, we

implemented fully factorized Gaussian variational posteriors to simplify computations of the evidence lower bound, following the Minimum Description Length approach. Besides, we employed the re-parametrization method, achieving a weight perturbation procedure supporting exploration of most relevant parts of the parameter space, averaged out by the EPF.

We evaluated the proposed EPF method on two markets to demonstrate both achieved performances and reusability in different conditions, namely Italian and Belgian markets. To enable reproducibility of our results, we exploited only data freely available from the Entso-e Transparency Platorm and GME/Terna websites, ranging from January 2015 to November 2018. Moreover, we included a quantitative analysis of major characteristics of the Italian PUN price since it was not available within the available research literature. Due to the lack of commonly adopted industry standard for EPF accuracy evaluation, we included several indicators commonly adopted within spot market field. Then, we demonstrated the capability of proposed method to achieve robust results in out-of-sample conditions in terms of forecast accuracy. Moreover, compared to conventional point forecast methods, the predictive distributions foster enhanced decision making, including planning of multiple strategies for the range of possible prices outcomes and short-term risk management.

Actually, we envision our study as a first step towards the full exploration of Bayesian deep learning for EPF. Indeed, several future extensions are foreseen. In particular, we plan to examine alternative priors and posteriors configurations within the variational inference framework. Afterwards, algorithms based on full posterior space search might be investigated to compare empirically related convergences, parametrization efforts and accuracy. Moreover, we plan to apply the Bayesian EPF framework to alternative deep neural network architectures, including recurrent neural network with Long short-term memory units, Convolutional neural networks and auto-encoders. Furthermore, we foresee the application of the proposed techniques to further energy markets (e.g., Nord Pool) characterized by different behavior, seasonality and potential features to be investigated.

Acknowledgements

The authors gratefully acknowledge the European Commission, for its support of the Horizon 2020 SYMBIOPTIMA project (grant agreement No 680426) and the Italian Ministry of Economic Development, for its support via the Accordo di Programma CNR-MSE PT 2015, Gruppo Tematico D.3: Processi e macchinari industriali.

References

[1] Weron R., Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach, (2006), Wiley-Publishing, Print ISBN:9780470057537 |Online ISBN:9781118673362, |DOI:10.1002/9781118673362

[2] Wang, D., Luo, H., Grunder, O., Lin, Y., & Guo, H. (2017). Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and BP neural network optimized by firefly algorithm. Applied Energy, 190, 390–407. https://doi.org/10.1016/j.apenergy.2016.12.134.

[3] Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. International Journal of Forecasting, 30(4), 1030–1081. https://doi.org/10.1016/j.ijforecast.2014.08.008

[4] Gianfreda, A., & Grossi, L. (2012). Forecasting Italian electricity zonal prices with exogenous variables. Energy Economics, 34(6), 2228–2239. https://doi.org/10.1016/j.eneco.2012.06.024

[5] Bunn, D. W. (Ed.) (2004). Modelling prices in competitive electricity markets. Chichester: Wiley.

[6] Zhang, X., Hug, G., & Harjunkoski, I. (2017). Cost-Effective Scheduling of Steel Plants with Flexible EAFs. IEEE Transactions on Smart Grid, 8(1), 239–249. https://doi.org/10.1109/TSG.2016.2575000

[7] Mitra, S., Sun, L., & Grossmann, I. E. (2013). Optimal scheduling of industrial combined heat and power plants under time-sensitive electricity prices. Energy, 54, 194–211. https://doi.org/10.1016/j.energy.2013.02.030

[8] Catalão JPS, Mariano SJPS, Mendes VMF, Ferreira LAFM. Short-term electricity prices forecasting in a competitive market: a neural network approach. Electr Power Syst Res 2007;77(10):1297–304.

[9] Mandal P, Srivastava AK, Senjyu T, Negnevitsky M. Electricity price forecasting using neural networks and similar days. Advances in Electric Power and Energy Systems Advances in Electric Power and Energy Systems: Load and Price Forecasting, 1st ed. John Wiley & Sons; 2017. Ch6, p. 215–50.

[10] Lago, J., De Ridder, F., Vrancx, P., & De Schutter, B. (2018). Forecasting day-ahead electricity prices in Europe: The importance of considering market integration. Applied Energy, 211(November 2017), 890–903. https://doi.org/10.1016/j.apenergy.2017.11.098

[11] D. Ramin, S. Spinelli, A. Brusaferri, Demand-side management via optimal production scheduling in power-intensive industries: the case of metal casting process, Applied Energy, 225 (2018), pp. 622-636

[12] Panapakidis, I. P., & Dagoumas, A. S. (2016). Day-ahead electricity price forecasting via the application of artificial neural network based models. Applied Energy, 172, 132–151. https://doi.org/10.1016/j.apenergy.2016.03.089

[13] Kaminski, V. (2013). Energy markets. Risk Books.

[14] Bento, P. M. R., Pombo, J. A. N., Calado, M. R. A., & Mariano, S. J. P. S. (2018). A bat optimized neural network and wavelet transform approach for short- term price forecasting. Applied Energy, 210(July 2017), 88–97. https://doi.org/10.1016/j.apenergy.2017.10.058

[15] Sharma, V., & Srinivasan, D. (2013). Engineering Applications of Artificial Intelligence A hybrid intelligent model based on recurrent neural networks and excitable dynamics for price prediction in deregulated electricity market. Engineering Applications of Artificial Intelligence, 26(5–6), 1562–1574. https://doi.org/10.1016/j.engappai.2012.12.012

[16] Ziel F, Steinert R, Husmann S. Efficient modeling and forecasting of electricity spot prices. Energy Econ 2015;47:98–111.

[17] A. J. Conejo, M. A. Plazas, R. Espinola, and A. B. Molina, "Day-ahead electricity price forecasting using the wavelet transform and ARIMA models," IEEE Trans. Power Syst., vol. 20, no. 2, pp. 1035–1042, May 2005.

[18] C. P. Rodriguez and G. J. Anders, "Energy price forecasting in the Ontario competitive power system market," IEEE Trans. Power Syst., vol. 19, no. 3, pp. 366–374, Feb. 2004.

[19] H. Y. Yamin, S. M. Shahidehpour, and Z. Li, "Adaptive short-term electricity price forecasting using artificial neural networks in the restructured power markets," International Journal of Electric Power, vol. 26, no. 8, pp. 571–581, Oct. 2004.

[20] Keles, D., Scelle, J., Paraschiv, F., & Fichtner, W. (2016). Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks. Applied Energy, 162, 218–230. https://doi.org/10.1016/j.apenergy.2015.09.087

[21] Nowotarski, J., & Weron, R. (2018). Recent advances in electricity price forecasting: A review of probabilistic forecasting. Renewable and Sustainable Energy Reviews, 81(September 2016), 1548–1568. https://doi.org/10.1016/j.rser.2017.05.234

[22] Sandhu, H. S., Fang, L., & Guan, L. (2016). Forecasting day-ahead price spikes for the Ontario electricity market. Electric Power Systems Research, 141, 450–459. https://doi.org/10.1016/j.epsr.2016.08.0057

[23] Gonzalez, V., Contreras, J., & Bunn, D. W. (2012). Forecasting power prices using a hybrid fundamental-econometric model. IEEE Transactions on Power Systems, 27(1), 363–372.

[24] Karakatsani, N. V., & Bunn, D. W. (2008). Forecasting electricity prices: the impact of fundamentals and time-varying coefficients. International Journal of Forecasting, 24(4), 764–785.

[25] Kristiansen, T. (2012). Forecasting Nord Pool day-ahead prices with an autoregressive model. Energy Policy, 49, 328–332.

[26] Liebl, D. (2013). Modeling and forecasting electricity spot prices: a functional data perspective. Annals of Applied Statistics, 7(3),1562–1592.

[27] Carmona, R., & Coulon, M. (2014). A survey of commodity markets and structural models for electricity prices. In F. E. Benth, V. Kholodnyi, and P. Laurence Ed., Quantitative energy finance: modeling, pricing, and hedging in energy and commodity markets. Springer.

[28] Cont, R., & Tankov, P. (2003). Financial modelling with jump processes, Chapman & Hall / CRC Press.

[29] Janczura, J., & Weron, R. (2010). An empirical comparison of alternate regime-switching models for electricity spot prices. Energy Economics, 32, 1059–1073.

[30] Weron, R., & Misiorek, A. (2008). Forecasting spot electricity prices: a comparison of parametric and semiparametric time series models. International Journal of Forecasting, 24, 744–763.

[31] Misiorek, A., Trück, S., & Weron, R. (2006). Point and interval forecasting of spot electricity prices: Linear vs. non-linear time series models. Studies in Nonlinear Dynamics and Econometrics, 10(3)

[32] Wood, A. J., & Wollenberg, B. F. (1996). Power generation, operation and control. New York: Wiley.

[33] Ventosa, M., Baíllo, Á., Ramos, A., & Rivier, M. (2005). Electricity market modeling trends. Energy Policy, 33(7), 897–913.

[34] Koritarov, V. S. (2004). Real-world market representation with agents. IEEE Power and Energy Magazine, 2(4), 39–46.

[35] Sousa, T. M., Pinto, T., Vale, Z., Praca, I., & Morais, H. (2012). Adaptive learning in multiagent systems: a forecasting methodology based on error analysis. Advances in Intelligent and Soft Computing, 156, 349–357.

[36] Ljung, L. (1999). System identification — theory for the user (2nd ed.). Prentice Hall: Upper Saddle River.

[37] Shumway, R. H., & Stoffer, D. S. (2006). Time series analysis and its applications (2nd ed.). Springer.

[38] Nogales, F. J., Contreras, J., Conejo, A. J., & Espinola, R. (2002). Forecasting next-day electricity prices by time series models. IEEE Transactions on Power Systems, 17, 342–348.

[39] Conejo, A. J., Contreras, J., Espinola, R., & Plazas, M. A. (2005). Forecasting electricity prices for a day-ahead pool-based electric energy market. International Journal of Forecasting, 21(3), 435–462.

[40] Schmutz, A., & Elkuch, P. (2004). Electricity price forecasting: application and experience in the European power markets. In Proceedings of the 6th IAEE European Conference, Zürich.

[41] Koopman, S. J., Ooms, M., & Carnero, M. A. (2007). Periodic seasonal reg-ARFIMA-GARCH models for daily electricity spot prices. Journal of the American Statistical Association, 102(477), 16–27.

[42] Karakatsani, N. V., & Bunn, D. W. (2010). Fundamental and behavioural drivers of electricity price volatility. Studies in Nonlinear Dynamics and Econometrics, 14(4), art. no. 4.

[43] Lin, T. N., Horne, B. G., Tino, P., & Giles, C. L. (1996). Learning long-term dependencies in NARX recurrent neural networks. IEEE Transactions on Neural Networks, 7(6), 1329–1337.

[44] Jonsson, T., Pinson, P., Nielsen, H. A., Madsen, H., & Nielsen, T. S. (2013). Forecasting electricity spot prices accounting for wind power predictions. IEEE Transactions on Sustainable Energy, 4(1), 210–218.

[45] Andalib, A., & Atry, F. (2009). Multi-step ahead forecasts for electricity prices using NARX: a new approach, a critical analysis of one-step ahead forecasts. Energy Conversion and Management, 50, 739–747.

[46] Aggarwal, S. K., Saini, L. M., & Kumar, A. (2009). Electricity price forecasting in deregulated markets: A review and evaluation. International Journal of Electrical Power and Energy Systems, 31, 13–22.

[47] Aggarwal, S. K., Saini, L. M., & Kumar, A. (2009). Short term price forecasting in deregulated electricity markets. A review of statistical models and key issues. International Journal of Energy Sector Management, 3(4), 333–358.

[48] Chen, X., Dong, Z. Y., Meng, K., Xu, Y., Wong, K. P., & Ngan, H. W. (2012). Electricity price forecasting with extreme learning machine and bootstrapping. IEEE Transactions on Power Systems, 27(4), 2055–2062.

[49] Cruz, A., Muñoz, A., Zamora, J. L., & Espinola, R. (2011). The effect of wind generation and weekday on Spanish electricity spot price forecasting. Electric Power Systems Research, 81(10), 1924–1935.

[50] Garcia-Ascanio, C., & Mate, C. (2010). Electric power demand forecasting using interval time series: A comparison between VAR and iMLP. Energy Policy, 38, 715–725.

[51] Gareta, R., Romeo, L. M., & Gil, A. (2006). Forecasting of electricity prices with neural networks. Energy Conversion and Management, 47, 1770–1778.

[52] Mandal, P., Senjyu, T., & Funabashi, T. (2006). Neural networks approach to forecast several hour ahead electricity prices and loads in deregulated market. Energy Conversion and Management, 47, 2128–2142.

[53] Pindoriya, N. M., Singh, S. N., & Singh, S. K. (2008). An adaptive wavelet neural network-based energy price forecasting in electricity markets. IEEE Transactions on Power Systems, 23, 1423–1432.

[54] Keynia, F., & Amjady, N. (2008). Electricity price forecasting with a new feature selection algorithm. Journal of Energy Markets, 1, 47–63.

[55] Amjady, N., & Keynia, F. (2009). Day-ahead price forecasting of electricity markets by a new feature selection algorithm and cascaded neural network technique. Energy Conversion and Management, 50, 2976–2982.

[56] Anbazhagan, S., & Kumarappan, N. (2013). Day-ahead deregulated electricity market price forecasting using recurrent neural network. IEEE Systems Journal, 7, 866–872.

[57] Y. Bengio, Learning deep architecture for ai, (2009) Found. Trends Mach. Learn., vol. 2, pp. 1–127.

[58] Pascanu, R., Montufar, G., and Bengio, Y. (2013). On the number of response regions of deep feed forward networks with piece-wise linear activations. ArXiv:1312.6098.

[59] X. Li and X. Wu, Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition (2015), in IEEE International Conference on Acoustics, Speech and Signal Processing , Apr. 2015, pp. 4520–4524.

[60] Le Roux, N. and Bengio, Y. (2010). Deep belief networks are compact universal approximators. Neural Computation, 22(8), 2192–2207.

[61] Delalleau, O. and Bengio, Y. (2011). Shallow vs. deep sum-product networks. In NIPS.

[62] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxoutnetworks. In ICML'2013.

[63] Hinton, G., Deng, L., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine, 29(6), 82–97.

[64] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. Technical report, arXiv:1207.0580.

[65] J. Lago, F.D. Ridder, B.D. Schutter, Forecasting spot electricity prices: deep learning approaches and empirical comparison of traditional algorithms, Applied Energy, 221 (2018), pp. 386-405

[66] Nowotarski, J., Weron, R. Recent advances in electricity price forecasting: A review of probabilistic forecasting, (2018) Renewable and Sustainable Energy Reviews, Part 1 81, pp. 1548-1568.

[67] Dudek G., Multilayer perceptron for gefcom2014 probabilistic electricity price forecasting, Int J Forecast, 32 (3) (2016), pp. 1057-1060

[68] Chen X, Dong Z, Meng K, Xu Y, Wong K, Ngan H. Electricity price forecasting with extreme learning machine and bootstrapping. (2012), IEEE Trans Power Syst, 2012;27(4):2055–62.

[69] V.Vahidinasab, S. Jadid, Bayesian neural network model to predict day-ahead electricity prices (2010), European Transactions on Electrical Power, 20, pp.231-246, DOI: 10.1002/etep.316

[70] European Network of Transmission System Operators, https://www.entsoe.eu/

[71] GME-Gestore Mercati Energetici, http://www.mercatoelettrico.org

[72] S. Ioffe, C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning, pp. 448–456, 2015.

[73] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15:1929–1958, 2014.

[74] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra. Weight uncertainty in neural networks. In Proceedings of the 32nd International Conference on Machine Learning, pp. 1613–1622, 2015.

[75] A.Graves. Practical variational inference for neural networks. In Advances in Neural Information Processing Systems, pp. 2348–2356, 2011.

[76] Murphy, K. P. (2013). Machine learning : a probabilistic perspective. MIT Press. ISBN: 0262018020

[77] R.M Neal. Bayesian learning for neural networks. PhD thesis, University of Toronto, 1995.

[78] M. Welling, Y. W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In Proceedings of the 28th International Conference on Machine Learning, pp. 681–688, 2011.

[79] H. Wang, D.Yeung. Towards Bayesian Deep Learning: A Framework and Some Existing Methods. IEEE Trans. on Knowl. and Data Eng. 28, 12 (2016), 3395-3408. DOI: https://doi.org/10.1109/TKDE.2016.2606428

[80] G. Hinton, D.V. Camp. Keeping the neural networks simple by minimizing the description length of the weights. In Proceedings of the 16th Annual Conference On Learning Theory (COLT), pages 5–13. ACM, 1993.

[81] J. Paisley, D. M. Blei, M. I. Jordan. 2012. Variational Bayesian inference with stochastic search. In Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML'12). Omnipress, USA, 1363-1370.

[82] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudoindependent weight perturbations on mini-batches. arXiv preprint arXiv:1803.04386, 2018.

[83] Kingma, D. P. & Ba, J. (2014). Adam: A Method for Stochastic Optimization.. CoRR, abs/1412.6980.

[84] Chen, D., & Bunn, D. W. (2010). Analysis of the nonlinear response of electricity prices to fundamental and strategic factors. IEEE Transactions on Power Systems, 25, 595–606.

[85] C.M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, Inc., New York, NY, USA, 1995.

[86] M. Lázaro-Gredilla and A. R. Figueiras-Vidal. Marginalized neural network mixtures for large-scale regression.IEEE transactions on neural networks, 21(8):1345–1351, 2010.

[87] R. Calandra, J. Peters, C. E. Rasmussen, M. P. Deisenroth. Manifold gaussian processes for regression. arXiv preprint arXiv:1402.5876, 2014.

[88] Goodfellow I, Bengio Y, Courville A. Deep learning. (2016) MIT Press; www.deeplearningbook.org

[89] Cuaresma, J. C., Hlouskova, J., Kossmeier, S., & Obersteiner, M. (2004). Forecasting electricity spot prices using linear univariate time-series models. Applied Energy, 77, 87–106.

[90] Haldrup, N., & Nielsen, M. Ø. (2006). A regime switching long memory model for electricity prices. Journal of Econometrics, 135, 349–376.

[91] Weron, R., & Misiorek, A. (2005). Forecasting spot electricity prices with time series models. IEEE Conference Proceedings, pp. 133–141.

[92] Zareipour, H., Canizares, C. A., Bhattacharya, K., & Thomson, J. (2006). Application of public-domain market information to forecast Ontario's wholesale electricity prices. IEEE Transactions on Power Systems, 21(4), 1707–1717.

[93] Lira, F., Muñoz, C., Nuñez, F., & Cipriano, A. (2009). Short-term forecasting of electricity prices in the Colombian electricity market. IET Generation, Transmission and Distribution, 3(11), 980–986.

[94] Albanese, D., Riccadonna, S., Donati, C., & Franceschi, P. (2018). A practical tool for maximal information coefficient analysis. GigaScience, 7(4), 1-8.

[95] D. Albanese, M. Filosi, R. Visintainer, S. Riccadonna, G. Jurman, C. Furlanello. minerva and minepy: a C engine for the MINE suite and its R, Python and MATLAB wrappers. Bioinformatics (2013) 29(3): 407-408 first published online December 14, 2012 doi:10.1093/bioinformatics/bts707.

[96] M.Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[97] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, R. A. Saurous.TensorFlow Distributions. arXiv preprint arXiv:1711.10604, 2017.

[98] C. Bergmeir and J. M. Bentez. On the use of cross-validation for time series predictor evaluation. Information Sciences, 191:192 – 213, 2012. Data Mining for Software Trustworthiness.

[99] P. Mandal, T. Senjyu, and T. Funabashi. Neural networks approach to forecast several hour ahead electricity prices and loads in deregulated market. Energy Conversion and Management, 47(15):2128 – 2142, 2006.

[100] H. Yamin, S. Shahidehpour, and Z. Li. Adaptive short-term electricity price forecasting using artificial neural networks in the restructured power markets. International Journal of Electrical Power and Energy Systems, 26:571 – 581, 2004.

[101] D. Hafner, D. Tran, T. Lillicrap, A. Irpan, J. Davidson, Reliable Uncertainty Estimates in Deep Neural Networks using Noise Contrastive Priors, 2018, arXiv:1807.09289

[102] Ben Taieb, Souhaib & Huser, Raphael & Hyndman, Rob & Genton, Marc. (2016). Forecasting Uncertainty in Electricity Smart Meter Data

by Boosting Additive Quantile Regression. IEEE Transactions on Smart Grid. 7. 1-8. 10.1109/TSG.2016.2527820.

[103] Ignacio Diaz-Emparanza. 2014. Numerical distribution functions for seasonal unit root tests. Comput. Stat. Data Anal. 76, C (August 2014), 237-247. DOI=http://dx.doi.org/10.1016/j.csda.2013.03.006

[104] R. Andrade, José & Filipe, Jorge & Reis, Marisa & J. Bessa, Ricardo. (2017). Probabilistic Price Forecasting for Day-Ahead and Intraday Markets: Beyond the Statistical Model. Sustainability. 9. 1990. 10.3390/su9111990.

[105] H. Hadera, I. Harjunkoski, G. Sand, I. E. Grossmann, S. Engell, Optimization of steel production scheduling with complex time-sensitive electricity cost, Computers & Chemical Engineering, Volume 76, 2015, Pages 117-136, ISSN 0098-1354, https://doi.org/10.1016/j.compchemeng.2015.02.004.

[106] gretl, Gnu Regression, Econometrics and Time-series Library, http://gretl.sourceforge.net/

[107] Morales, Juan & J. Conejo, Antonio & Madsen, Henrik & Pinson, Pierre & Zugno, Marco. (2014). Integrating Renewables in Electricity Markets - Operational Problems., Springer, ISBN 978-1-4614-9411-9

[108] U. Ugurlu, I. Oksuz, O. Tas, (2018). Electricity Price Forecasting Using Recurrent Neural Networks. Energies. 11. 1255. 10.3390/en11051255.

[109] R. Beigaitė, T. Krilavičius, Electricity price forecasting for Nord Pool data using recurrent neural networks, CEUR Workshop proceedings: IVUS 2018, International conference on information technologies, Kaunas, Lithuania, 27 April, 2018. Aachen : CEUR-WS, 2018, Vol. 2145

[110] L. Peng, S. Liu, R. Liu, L. Wang, Effective long short-term memory with differential evolution algorithm for electricity price prediction, Energy, Volume 162, 2018, Pages 1301-1314, ISSN 0360-5442, https://doi.org/10.1016/j.energy.2018.05.052.

[111] S. Bordignon, D. W. Bunn, F. Lisi, F. Nan, Combining day-ahead forecasts for British electricity prices, Energy Economics, Volume 35, 2013, Pages 88-103, ISSN 0140-9883, https://doi.org/10.1016/j.eneco.2011.12.001.

[112] J. Nowotarski, E. Raviv, S. Trück, R. Weron, An empirical comparison of alternative schemes for combining electricity spot price forecasts, Energy Economics, Volume 46, 2014, Pages 395-412, ISSN 0140-9883, https://doi.org/10.1016/j.eneco.2014.07.014.

[113] S. Mitra, L. Sun, I. E. Grossmann, Optimal scheduling of industrial combined heat and power plants under time-sensitive electricity prices, Energy, Volume 54, 2013, Pages 194-211, ISSN 0360-5442, https://doi.org/10.1016/j.energy.2013.02.030.

Appendix

Table A. Descriptive statistics of working days

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Mean	46.3	42.3	39.7	38.3	38.5	41.6	49.0	56.2	62.7	60.5	56.1	53.7	48.9	47.9	51.0	54.0	57.3	61.0	63.7	65.8	63.6	58.4	53.5	49.2
Std	9.8	9.3	9.2	9.0	8.7	8.5	10.1	13.5	16.5	16.0	14.4	13.9	12.7	13.1	14.5	15.3	16.6	19.7	19.4	17.3	13.3	11.3	9.8	9.3
Min	20.9	18.5	14.9	11.1	12.1	13.4	26.3	30.8	30.2	30.3	23.6	20.0	14.1	6.1	6.9	8.4	9.8	16.6	27.9	31.5	33.7	33.7	30.5	25.8
Max	40.3	36.7	34.0	32.4	33.0	36.0	42.6	48.4	52.0	50.5	47.3	45.1	40.8	39.5	42.1	44.3	46.8	48.0	50.6	54.3	55.3	52.5	48.5	44.0
25%	45.8	42.3	39.9	38.3	38.5	41.2	48.3	54.2	60.7	57.8	53.7	51.9	48.2	47.1	49.7	52.1	54.7	56.2	58.7	61.9	61.9	57.6	53.1	49.5
50%	50.9	47.0	44.5	43.1	43.1	46.1	54.6	62.7	70.7	68.4	62.9	60.5	55.3	54.6	58.5	62.0	65.7	70.1	75.4	75.5	71.1	63.2	57.6	53.3
75%	111.9	111.1	109.1	96.2	95.9	96.2	110.3	139.9	162.4	160.7	140.0	132.3	126.1	135.0	140.7	140.4	164.3	170.0	165.1	160.0	145.7	137.3	135.2	130.8

Table B. Descriptive statistics of non-working days

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Mean	47.6	43.3	40.3	38.3	37.8	39.0	40.6	43.1	45.2	46.2	45.0	42.4	39.8	36.0	35.7	38.5	43.0	48.5	54.1	59.3	59.4	55.2	50.5	46.3
Std	8.8	8.6	8.9	8.9	9.1	9.0	9.3	10.3	10.8	11.0	10.6	10.8	10.9	11.6	12.0	11.7	11.6	13.7	13.5	13.9	11.8	9.8	8.2	7.6
Min	26.0	17.9	10.8	5.0	3.3	5.0	8.3	9.4	14.3	14.4	12.7	10.3	7.4	2.2	3.0	6.4	8.5	12.7	25.0	28.6	30.9	28.3	28.4	25.0
Max	41.7	37.6	35.0	33.0	32.8	34.0	34.7	36.4	38.2	39.1	38.4	35.9	33.0	29.0	28.5	31.5	35.5	39.5	44.5	50.1	51.7	49.0	45.0	41.2
25%	46.9	43.8	40.5	39.2	38.6	39.3	40.5	43.2	45.3	46.4	45.2	42.5	40.3	36.8	36.3	38.4	42.5	46.9	52.3	57.5	57.9	54.9	50.7	46.8
50%	53.3	48.6	45.7	44.1	43.8	45.0	46.4	49.4	51.6	52.2	51.6	49.3	46.3	43.4	43.6	46.9	50.6	56.0	63.9	69.1	66.3	61.2	55.6	51.5
75%	93.1	75.5	68.0	65.3	65.3	69.0	67.4	83.0	87.2	87.8	81.5	78.6	68.6	63.5	68.7	76.3	80.0	99.9	93.4	112.4	111.4	82.4	75.0	69.1

Table 8. Italian market test set performances on different weekdays

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
RMSE BNN	9.6	9.8	9.4	9.2	9.4	8.1	7.4
RMSE DNN	10.0	8.7	8.8	8.8	9.7	8.7	8.0
MAPE BNN	12.0	11.5	11.1	10.7	10.9	11.1	12.2
MAPE DNN	12.7	10.6	10.5	10.9	11.0	11.9	12.9
SMAPE BNN	12.0	11.5	11.4	10.8	11.1	11.7	11.8
SMAPE DNN	12.9	10.5	10.7	10.9	11.3	12.7	12.8

Table 9. Belgian market test set performances on different weekdays

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
RMSE BNN	10.4	14.2	10.4	10.7	11.2	11.3	10.4
RMSE DNN	10.8	14.0	11.2	11.0	11.1	11.5	10.4
MAPE BNN	16.1	17.2	13.5	13.9	16.0	15.8	23.7
MAPE DNN	17.6	16.9	13.5	13.8	15.8	15.6	23.5
SMAPE BNN	18.0	17.0	13.3	14.1	13.5	16.3	20.9
SMAPE DNN	18.9	17.0	13.5	14.1	13.7	16.1	20.4

Table 10. Italian market test set performances on different months

	Jan	Feb	Mar	Apr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dec
RMSE BNN	5.9	8.6	9.9	7.8	6.7	9.0	11.0	9.9	11.4	10.0	8.8	8.8
RMSE DNN	7.2	10.4	9.7	8.6	6.9	8.8	7.7	10.3	10.3	11.1	8.3	7.2
MAPE BNN	10.0	9.4	12.5	13.8	10.5	13.8	13.1	11.6	11.0	11.5	9.0	9.9
MAPE DNN	12.3	12.2	11.6	14.6	10.7	13.1	9.5	12.7	10.1	13.0	8.2	12.3
SMAPE BNN	9.8	9.5	11.8	12.9	10.6	15.1	12.1	12.0	11.9	12.4	9.4	10.3
SMAPE DNN	11.4	11.8	11.6	14.2	10.8	14.3	9.5	13.5	10.9	14.3	8.5	11.4

Table 11. Belgian market test set performances on different months

	Jan	Feb	Mar	Apr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dec
RMSE BNN	7.5	6.7	12.9	8.4	12.9	12.1	7.3	8.6	12.5	15.1	13.2	14.6
RMSE DNN	7.4	7.1	12.3	8.9	13.1	11.3	8.1	9.3	12.2	16.8	13.0	14.4
MAPE BNN	23.2	10.3	14.3	23.9	25.1	20.0	10.7	10.8	13.2	14.3	15.2	17.4
MAPE DNN	20.3	11.0	14.0	25.1	25.4	18.1	11.6	11.2	12.9	14.7	15.4	18.0
SMAPE BNN	20.0	10.3	14.7	18.5	25.2	21.0	11.1	11.3	13.2	14.8	14.8	16.6
SMAPE DNN	19.5	10.8	14.4	18.8	25.8	18.5	12.0	11.7	13.2	15.8	15.1	17.0