Published in Solar Energy, 2021, vol. 225, pp. 577-590,
which should be cited to refer to this work.

DOI: https://doi.org/10.1016/j.solener.2021.07.063

Bayesian bootstrapping in real-time probabilistic photovoltaic power forecasting

Mokhtar Bozorg [a], Antonio Bracale [b], Mauro Carpita [a], Pasquale De Falco [b,*], Fabio Mottola [c], and Daniela Proto [c]

[a] *Institute of Energy and Electrical Systems, University of Applied Sciences Western Switzerland (HES-SO), Yverdon-les-Bains 1401, Switzerland. E-mails: mokhtar.bozorg@heig-vd.ch; mauro.carpita@heig-vd.ch*

[b] *Department of Engineering, University of Naples Parthenope, Naples 80143, Italy. E-mails: antonio.bracale@uniparthenope.it; pasquale.defalco@uniparthenope.it*

[c] *Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples 80125, Italy. Emails: fabio.mottola@unina.it; daniela.proto@unina.it*

[*] *Corresponding author*

Abstract

Modern distribution systems are characterized by increasing penetration of photovoltaic generation systems. Due to the uncertain nature of the solar primary source, photovoltaic power forecasting models must be developed in any energy management system for smart distribution networks. Although point forecasts can suit many scopes, probabilistic forecasts add further flexibility to any energy management system, and they are recommended to enable a wider range of decision making and optimization strategies. Real-time probabilistic photovoltaic power forecasting is performed in this paper by using an approach based on Bayesian bootstrap. Particularly, the Bayesian bootstrap is applied to three probabilistic forecasting models (i.e., linear quantile regression, gradient boosting regression tree and quantile regression neural network) to provide sample bootstrap distributions of the predictive quantiles of photovoltaic power. The heterogeneous nature of the selected models allows evaluating the performance of the Bayesian bootstrap within different forecasting frameworks. Several benchmarks and error indices and scores are used to assess the performance of Bayesian bootstrap in probabilistic photovoltaic power forecasting. Tests carried out on an actual photovoltaic power dataset for probabilistic forecasting demonstrates the effectiveness of the proposed approach.

Keywords: Bayesian bootstrap; photovoltaic power forecasting; probabilistic forecasting; renewable energy.

1. Introduction

Management of smart grids requires strategies based on accurate forecasting of loads and energy resources. Increase of production by renewable generation systems makes forecasting of available energy a complex challenge due to the randomness of the primary energy source (typically, solar and wind). Uncertainties are even more critical in real-time operation which requires more accurate forecasting to avoid insecure, inefficient or even unfeasible operation. Errors in real-time forecasting, in fact, would lead to incorrect operation of the grid as for instance by the undesired intervention of frequency control resources to compensate load/generation imbalances. In addition, the rapid dynamic of modern network operating conditions makes real-time optimal operation a mandatory issue. Fortunately, short time frames can benefit from more accurate information provided by local sensor devices, which are able to update the prediction with current values of the variables being forecasted. This is

particularly suitable when dealing with statistical learning models which can be trained by updated data based on current conditions of the quantity of interest.

Most of the real-time forecasting approaches are based on point forecast methods which give single quantities as a result of the forecast. These methods, however, have reduced utility in power system operation and management as, for instance, they have minimal usefulness in solving risk-based tasks such as market bidding (Van der Meer et al., 2018, Bessa et al., 2017). The intrinsic randomness of the energy source, in addition to the uncertainties associated with intra-day and real-time market operations, makes probabilistic approaches more suitable. Probabilistic approaches produce forecasts in the form of quantiles, intervals or density functions, which provide more comprehensive information compared to point forecasts (Van der Meer et al., 2018). Nevertheless, the technical literature has started dealing with probabilistic approaches only in the last years.

The methods typically used in the probabilistic framework are statistical models or hybrid physical-statistical models (i.e., coupling weather models and statistical approaches). By focusing on forecast models of solar irradiance and photovoltaic (PV) power production, Linear Quantile Regression (LQR) models are proposed in [3,4]. The LQR approach is used in (Juban et al., 2016) to predict the distribution of outcomes. Probabilistic models for intra-day solar forecasting are carried out in the case of highly variable sky conditions based on an LQR model in (Lauret et al., 2017). Other approaches are based on machine learning techniques (Voyant et al., 2017). The performance of machine learning methods (k-nearest-neighbors and gradient boosting regression trees (GBRT)) for both deterministic and probabilistic intra-hour forecasting of solar irradiance is evaluated in (Pedro et al., 2018). Applications of regression trees to solar forecasting are presented in (Voyant et al., 2018). A hybrid physical-statistical model developed in (Almeida et al., 2015) uses Quantile Regression Forests (QRF) to forecast power output employing as inputs predicted meteorological variables from a Numeric Weather Prediction (NWP) model and actual power measurements of PV plants. Quantile Regression Neural Network (QRNN) is used in (Fernandez-Jimenez et al., 2017) for probabilistic short-term forecasting of the power production in a PV power plant. A real-time hybrid probabilistic model is developed for intra-hour horizons in (Chu et al., 2015).

It is worth to note that it is crucial to improve predictions of some weather variables through the proper model selection, which allows selecting the most informative predictors and discarding uninformative inputs. To do that the exploitation of available input data must be maximized. In probabilistic energy forecasting, this has been successfully done through ensemble approaches (Ren et al., 2015, La Salle et al., 2020), as evidenced also by the results of the PV forecasting track of the Global Energy Forecasting Competition 2014 (Hong et al., 2016). Among these, widely applicable and powerful tools are those based on the bootstrap methodology.

The bootstrap is a statistical tool which can be tailored to quantify the uncertainty related to a specific estimator or statistical learning method. In the bootstrap method, samples are repeatedly extracted with replacement from the available data. For each of these set of samples, models are estimated and used to generate differentiated forecasts. Regarding the suitability of this technique for real-time applications, it must be noted that despite bootstrap can appear characterized by a heavy computational burden, it is a flexible technique which can be applied on some selected suitable parameters, thus reducing the computational time. Focusing on real-time application, the bootstrap technique has been used to monitor PV power plant output in (Vergura et al., 2009). A two-stage method is proposed in (Wen et al., 2020) to estimate the uncertainty of PV forecasting; the forecasting is based on the bootstrap method applied to short-term deterministic predictions obtained from a hybrid intelligent model that combines wavelet transform technique for data pre-processing, and radial basis function neural network method. Within the Bayesian framework – which is successfully applied to short-term photovoltaic forecasting, e.g., (Doubleday et al., 2020) and

(Lee et al., 2019) – bootstrapping techniques have also been used to improve the probabilistic forecasts. Bayesian bootstrap techniques for short-term PV power forecasting have been, however, rarely adopted (Bozorg et al., 2020).

This paper focuses on the application of Bayesian bootstrap in short-term probabilistic PV forecasting. The Bayesian bootstrap is specifically suited up in this paper to be applied to three different underlying probabilistic models (i.e., LQR, GBRT and QRNN), in order to evaluate potential improvements due to its application. The major aim of this research is indeed to evaluate if the Bayesian bootstrap enables for better performance and increased skill of the forecasts, compared to the stand-alone usage of the underlying probabilistic models and compared to the application of the traditional bootstrap. Another contribution of this research is the development of the three aforementioned probabilistic forecasting methods which are developed for the first time under a new framework. Compared to (Bozorg et al., 2020), the three methods make the Bayesian bootstrap operate directly on the PV power forecasts, rather than on the parameters of the models. This approach therefore allows reducing the overall computational effort which is particularly important in short-term forecasting, thanks to the fact that there is no need to pass through the sample bootstrap distributions of the parameters since the sample Bayesian bootstrap distribution of the predictive quantile of PV power is directly provided.

Eventually, a dedicated procedure for the extraction of the optimal sample quantile from the sample Bayesian bootstrap distribution is also proposed, in order to suit the requirements of common decision-making tools that accept probabilistic forecasts as a set of predictive quantiles.

This paper is organized as follows. The Bayesian bootstrap analytic details are presented in Section 2. The application of the Bayesian bootstrap to probabilistic models for short-term PV power forecasting is described in Section 3. The background for the forecasting performance assessment is shown in Section 4, whereas Section 5 presents the results of numerical experiments based on actual PV data. The paper is concluded in Section 6.

2. The Bayesian bootstrap

Let's assume to be interested in characterizing a target statistic $\varphi(\boldsymbol{x}) \in \mathbb{R}^s$ that is a function of a (row) vector $\boldsymbol{x} = \{x_1, \dots, x_{N_b}\}$ of N_b variables, and let's assume that an estimation $\hat{\varphi}(\boldsymbol{x})$ of the target, statistic can be characterized using an available dataset $\boldsymbol{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_M\}$ that contains M known occurrences of $\boldsymbol{x}$.

Bootstrapping is a resampling approach that allows estimating the probabilistic properties of the target statistic by randomly sampling with replacement from data $\boldsymbol{X}$ (Hastie et al., 2009, Clyde et al., 2001). Under the previous assumptions, $\boldsymbol{X}$ is a $M \times N_b$ matrix and its generic m^{th} row is $\boldsymbol{x}_m = \{x_{m,1}, \dots, x_{m,N_b}\}$. The M rows of $\boldsymbol{X}$ can theoretically be assumed as M elements extracted from an unknown N_b-variate distribution $F(\boldsymbol{x}) \in \mathcal{F}$, and within a probability distribution framework the estimated target statistic $\hat{\varphi}(\boldsymbol{x})$ retains the statistical properties of a function $G[\cdot]$ applied to $F(\boldsymbol{x})$, i.e., $G[F(\boldsymbol{x})]$, that maps from $\mathcal{F}$ to $\mathbb{R}^s$.

Bootstrapping estimates the target statistic in terms of an empirical R -sample bootstrap distribution $G\left[F^{\langle 1\rangle}(\boldsymbol{x})\right], \dots, G[F^{\langle R\rangle}(\boldsymbol{x})]$, obtained by applying the function $G[\cdot]$ to R replicates $F^{\langle 1\rangle}(\boldsymbol{x}), \dots, F^{\langle R\rangle}(\boldsymbol{x})$. Specifically, bootstrapping assumes the type of the unknown distribution $F(\boldsymbol{x})$ to be:

$$F(\boldsymbol{x}) = \sum_{m=1}^{M} w_m \cdot \delta_{\boldsymbol{x}_m}, \qquad \sum_{m=1}^{M} w_m = 1 \text{ and } w_m \geq 0, \tag{1}$$

where w_m is a weight assigned to the m^{th} occurrence, and $\delta_{\boldsymbol{x}_m}$ is a degenerate probability measure for the m^{th} vector $\boldsymbol{x}_m$.

The generic r^{th} replicate is:

$$F^{\langle r\rangle}(\boldsymbol{x}) = \sum_{m=1}^{M} w_m^{\langle r\rangle} \cdot \delta_{\boldsymbol{x}_m}, \qquad \sum_{m=1}^{M} w_m^{\langle r\rangle} = 1 \text{ and } w_m^{\langle r\rangle} \geq 0, \tag{2}$$

where the weights $\boldsymbol{w}^{\langle r\rangle} = \{w_1^{\langle r\rangle}, \dots, w_M^{\langle r\rangle}\}$ are extracted randomly from an assigned M-variate distribution $f_{\boldsymbol{w}}$.

In traditional bootstrap, the distribution $f_{\boldsymbol{w}}$ from which the weights $\boldsymbol{w}^{\langle r\rangle}$ are extracted is the multinomial distribution in M dimensions with equal probabilities $1/M$, i.e.:

$$f_{\boldsymbol{w}} = \mathrm{Mul}(w_1, \dots, w_M | M; 1/M, 1/M, \dots, 1/M) = \frac{M!}{w_1! \dots w_M!} \prod_{m=1}^{M} \left(\frac{1}{M}\right)^{w_m}; \tag{3}$$

the M-variate sample extracted from (3) is then normalized by M (i.e., $w_m^{\langle r\rangle} = w_m/M$), to meet the constraint $\sum_{m=1}^{M} w_m^{\langle r\rangle} = 1$.

In Bayesian bootstrap, instead, the weights $\boldsymbol{w} = \{w_1, \dots, w_M\}$ are estimated through the Bayesian inference upon the observed probabilities $\widehat{\boldsymbol{w}} = \{\widehat{w}_1, \dots, \widehat{w}_M\}$ of data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M$ in $\boldsymbol{X}$. The posterior distribution $p(\boldsymbol{w}|\widehat{\boldsymbol{w}}, \boldsymbol{\alpha})$ of the weights is obtainable by assigning a prior distribution $p(\boldsymbol{w}|\boldsymbol{\alpha})$ to the objective weights $\boldsymbol{w}$, which has a general set of parameters $\boldsymbol{\alpha}$. To allow the calculation of the posterior distribution in closed form, the prior distribution is selected as a conjugate prior of the likelihood. Since the likelihood is multinomial (Hastie et al., 2009, Clyde et al., 2001, Rubin, 1981), the prior distribution is a symmetric Dirichlet that has all parameters $\boldsymbol{\alpha}$ equal to α, i.e., $\boldsymbol{\alpha} = \{\alpha, \dots, \alpha\}$:

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \mathrm{Dir}(w_1, \dots, w_M | \alpha, \dots, \alpha) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \cdot \prod_{m=1}^{M} w_m^{(\alpha-1)}, \tag{4}$$

where $\mathrm{B}(\boldsymbol{\alpha})$ is the Beta function calculated on the M-dimensional vector $\boldsymbol{\alpha} = \{\alpha, \dots, \alpha\}$. With this position, the corresponding posterior distribution is a Dirichlet too:

$$p(\boldsymbol{w}|\widehat{\boldsymbol{w}}, \boldsymbol{\alpha}) = \mathrm{Dir}(w_1, \dots, w_M | \alpha + M\widehat{w}_1, \dots, \alpha + M\widehat{w}_M) = \frac{1}{\mathrm{B}(\alpha + M\widehat{w}_1, \dots, \alpha + M\widehat{w}_M)} \cdot \prod_{m=1}^{M} w_m^{(\alpha + M\widehat{w}_m - 1)}. \tag{5}$$

If the prior is uninformative, as in (Hastie et al., 2009, Clyde et al., 2001, Rubin, 1981), $\alpha = 0$ and:

$$p(\boldsymbol{w}|\widehat{\boldsymbol{w}}, \boldsymbol{0}) = \mathrm{Dir}(w_1, \dots, w_M | M\widehat{w}_1, \dots, M\widehat{w}_M) = \frac{1}{\mathrm{B}(M\widehat{w}_1, \dots, M\widehat{w}_M)} \cdot \prod_{m=1}^{M} w_m^{(M\widehat{w}_m - 1)}, \tag{6}$$

which, for continuous variables, can be further and reasonably written as $\mathrm{Dir}(w_1, \dots, w_M | 1, \dots, 1)$ since the observed rows $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M$ likely have the probability $1/M$ to occur (i.e., only once in the entire dataset $\boldsymbol{X}$) (Rubin et al., 1981).

In Bayesian bootstrap, in summary, the distribution $f_{\boldsymbol{w}}$ from which the weights $\boldsymbol{w}^{\langle r\rangle}$ are directly extracted is the M-variate flat Dirichlet distribution, i.e.:

$$f_{\boldsymbol{w}} = \mathrm{Dir}(w_1, \dots, w_M | 1, \dots, 1). \tag{7}$$

Once the weights are extracted, the function $G[\cdot]$ is applied to $F^{\langle 1\rangle}(\boldsymbol{x}), \dots, F^{\langle R\rangle}(\boldsymbol{x})$ given by (3), and the R-sample Bayesian bootstrap distribution $G\left[F^{\langle 1\rangle}(\boldsymbol{x})\right], \dots, G[F^{\langle R\rangle}(\boldsymbol{x})]$ is the output of the Bayesian bootstrap that gives an estimation $\hat{\varphi}(\boldsymbol{x})$ of the target statistic.

3. Applications of the Bayesian bootstrap to probabilistic models for PV power forecasting

Traditional bootstrap is commonly applied in forecasting problems to generate probabilistic predictions or to characterize the uncertainty of predictive parameters (Hastie et al., 2009). In this paper, the Bayesian bootstrap is applied to three probabilistic forecasting models (LQR, GBRT and QRNN) to provide sample bootstrap distributions of the predictive quantiles of PV power. The models, selected from the literature, are purposely very heterogeneous by nature to evaluate the performance of the Bayesian bootstrap within different forecasting frameworks.

The three probabilistic forecasting models are briefly recalled in the first part of this Section; then the role of Bayesian bootstrap in the forecasting system and a procedure to optimize the Bayesian-bootstrap-based predictions are detailed in the second part of this Section.

3.1. Linear Quantile Regression model

An LQR model allows estimating the cumulative distribution function of the PV power by means of a linear relationship between the power and informative predictors (Juban et al., 2016). The linear relationship is established between the predictive α_q -quantile $\hat{P}_h^{\langle \alpha_q\rangle}$ of PV power and the predictors $\boldsymbol{z}_h = \{z_{h,1}, \dots, z_{h,N}\}$ through a set of model parameters $\boldsymbol{\beta}^{\langle \alpha_q\rangle} = \left\{\beta_0^{\langle \alpha_q\rangle}, \dots, \beta_N^{\langle \alpha_q\rangle}\right\}$ which have to be estimated. Both quantile and predictors refer to the time horizon h, and the forecast lead time k is assumed for predictors, i.e., their values are available at $h-k$, that is the forecast origin. For ease of notation, in what follows the forecast lead time k is not included in the symbols. The linear relationship imposed by a generic QR model is provided by:

$$P_h^{\langle \alpha_q\rangle}\left(\boldsymbol{z}_h \mid \boldsymbol{\beta}^{\langle \alpha_q\rangle}\right) = \beta_0^{\langle \alpha_q\rangle} + \sum_{n=1}^{N} \beta_n^{\langle \alpha_q\rangle} \cdot z_{h,n}\,, \tag{8}$$

where the vector $\boldsymbol{\beta}^{\langle \alpha_q\rangle} = \left\{\beta_0^{\langle \alpha_q\rangle}, \dots, \beta_N^{\langle \alpha_q\rangle}\right\}$ includes the $N+1$ parameters of the LQR model. The estimated values $\widehat{\boldsymbol{\beta}}^{\langle \alpha_q\rangle}$ of $\boldsymbol{\beta}^{\langle \alpha_q\rangle}$ derive from the model training obtained by the minimization of a proper score evaluated respect to a known data set (i.e., the supervised training set that includes M_{tr} training PV power samples $\boldsymbol{P}^{\langle tr\rangle} = \{P_t, t \in \Omega^{\langle tr\rangle}\}$ and the corresponding predictors $\boldsymbol{Z}^{\langle tr\rangle} = \{\boldsymbol{z}_t, t \in \Omega^{\langle tr\rangle}\}$. The Pinball Score (PS) is used for this purpose. PS is a score calculated on the training samples $\boldsymbol{P}^{\langle tr\rangle}$ and on the corresponding M_{tr} α_q-quantiles $\boldsymbol{P}^{\langle tr\rangle,\langle \alpha_q\rangle}\left(\boldsymbol{Z}^{\langle tr\rangle} \mid \boldsymbol{\beta}^{\langle \alpha_q\rangle}\right) = \left\{P_t^{\langle \alpha_q\rangle}\left(\boldsymbol{z}_t \mid \boldsymbol{\beta}^{\langle \alpha_q\rangle}\right), t \in \Omega^{\langle tr\rangle}\right\}$ given by the LQR model:

$$\begin{aligned} &PS\left[\boldsymbol{P}^{\langle tr\rangle}, \boldsymbol{P}^{\langle tr\rangle,\langle \alpha_q\rangle}\left(\boldsymbol{Z}^{\langle tr\rangle} \mid \boldsymbol{\beta}^{\langle \alpha_q\rangle}\right)\right] = \tfrac{1}{M_{tr}} \sum_{t\in\Omega^{\langle tr\rangle}} PS\left[P_t, P_t^{\langle \alpha_q\rangle}\left(\boldsymbol{z}_t \mid \boldsymbol{\beta}^{\langle \alpha_q\rangle}\right)\right] = \\ &= \tfrac{1}{M_{tr}} \sum_{t\in\Omega^{\langle tr\rangle}} \left\{\alpha_q - \mathrm{I}\left[P_t \le P_t^{\langle \alpha_q\rangle}\left(\boldsymbol{z}_t \mid \boldsymbol{\beta}^{\langle \alpha_q\rangle}\right)\right]\right\} \cdot \left[P_t - P_t^{\langle \alpha_q\rangle}\left(\boldsymbol{z}_t \mid \boldsymbol{\beta}^{\langle \alpha_q\rangle}\right)\right], \end{aligned} \tag{9}$$

where $\mathrm{I}\left[P_t \le P_t^{\langle \alpha_q\rangle}\left(\boldsymbol{z}_t \mid \boldsymbol{\beta}^{\langle \alpha_q\rangle}\right)\right]$ is the indicator function that depends on the condition in the brackets:

$$\mathrm{I}\left[P_t \le P_t^{\langle\alpha_q\rangle}\left(\boldsymbol{z}_t \mid \boldsymbol{\beta}^{\langle\alpha_q\rangle}\right)\right] = \begin{cases} 1 & \text{if } P_t \le P_t^{\langle\alpha_q\rangle}\left(\boldsymbol{z}_t \mid \boldsymbol{\beta}^{\langle\alpha_q\rangle}\right) \\ 0 & \text{if } P_t > P_t^{\langle\alpha_q\rangle}\left(\boldsymbol{z}_t \mid \boldsymbol{\beta}^{\langle\alpha_q\rangle}\right). \end{cases} \tag{10}$$

The estimated parameter vector $\widehat{\boldsymbol{\beta}}^{\langle\alpha_q\rangle}$ is eventually given by:

$$\widehat{\boldsymbol{\beta}}^{\langle\alpha_q\rangle} = \underset{\boldsymbol{\beta}^{\langle\alpha_q\rangle}}{\operatorname{argmin}}\ PS\left[\boldsymbol{P}^{\langle tr\rangle}, \boldsymbol{P}^{\langle tr\rangle,\langle\alpha_q\rangle}\left(\boldsymbol{Z}^{\langle tr\rangle} \mid \boldsymbol{\beta}^{\langle\alpha_q\rangle}\right)\right], \tag{11}$$

i.e., it is a function of $\boldsymbol{P}^{\langle tr\rangle}$ and $\boldsymbol{Z}^{\langle tr\rangle}$:

$$\widehat{\boldsymbol{\beta}}^{\langle\alpha_q\rangle} = \widehat{\boldsymbol{\beta}}^{\langle\alpha_q\rangle}\left(\boldsymbol{P}^{\langle tr\rangle}, \boldsymbol{Z}^{\langle tr\rangle}\right). \tag{12}$$

The LQR predictive α_q-quantile $\hat{P}_h^{\langle\alpha_q\rangle}$ of PV power for the time horizon h, returned by using the estimated parameters, consequently depends on $\boldsymbol{P}^{\langle tr\rangle}$ and $\boldsymbol{Z}^{\langle tr\rangle}$, too:

$$\hat{P}_h^{\langle\alpha_q\rangle}\left[\boldsymbol{z}_h \mid \widehat{\boldsymbol{\beta}}^{\langle\alpha_q\rangle}\left(\boldsymbol{P}^{\langle tr\rangle}, \boldsymbol{Z}^{\langle tr\rangle}\right)\right] = \hat{\beta}_0^{\langle\alpha_q\rangle} + \textstyle\sum_{n=1}^{N} \hat{\beta}_n^{\langle\alpha_q\rangle} \cdot z_{h,n}. \tag{13}$$

3.2. Gradient Boosting Regression Trees

GBRTs focus on the functional dependence $f^{\langle\alpha_q\rangle}(\cdot)$ between the response variable (the predictive α_q-quantile of PV power) $P_h^{\langle\alpha_q\rangle}$ and the corresponding predictors $\boldsymbol{z}_h$ (Alexey and Alois, 2013, Friedman, 2002):

$$P_h^{\langle\alpha_q\rangle} = f^{\langle\alpha_q\rangle}(\boldsymbol{z}_h)\ . \tag{14}$$

Assuming $f^{\langle\alpha_q\rangle}$ to be unknown, the predictive α_q-quantile of PV power depends on the function $f^{\langle\alpha_q\rangle}$, i.e., $P_h^{\langle\alpha_q\rangle} = P_h^{\langle\alpha_q\rangle}(\boldsymbol{z}_h | f^{\langle\alpha_q\rangle})$. An estimation $\hat{f}^{\langle\alpha_q\rangle}$ of $f^{\langle\alpha_q\rangle}$ can be obtained by minimizing the PS over the training data (Alexey and Alois, 2013, Friedman, 2002):

$$\hat{f}^{\langle\alpha_q\rangle} = \underset{f^{\langle\alpha_q\rangle}}{\operatorname{argmin}}\ PS\left[\boldsymbol{P}^{\langle tr\rangle}, \boldsymbol{P}^{\langle tr\rangle,\langle\alpha_q\rangle}\left(\boldsymbol{Z}^{\langle tr\rangle} \mid f^{\langle\alpha_q\rangle}\right)\right]\ . \tag{15}$$

The iterative procedure for solving (15) starts at iteration $j = 0$ by initializing $\hat{f}_{\langle 0\rangle}$ at the constant value:

$$\hat{f}_{\langle 0\rangle}^{\langle\alpha_q\rangle} = \hat{\rho}_{\langle 0\rangle}^{\langle\alpha_q\rangle} = \underset{\rho}{\operatorname{argmin}} \textstyle\sum_{t\in\Omega^{\langle tr\rangle}} PS[P_t, \rho]. \tag{16}$$

The GBRT iterative procedure at the iteration $j > 0$ uses gradient descent to create new learners and the new estimation is made through the negative gradient, that is the negative partial derivative of the PS loss function evaluated for the m^{th} historical observation P_t:

$$g_t^{\langle \alpha_q \rangle}(\boldsymbol{z}_t) = -\left\{ \frac{\partial PS\left[P_t, f_{\langle j-1 \rangle}^{\langle \alpha_q \rangle}(\boldsymbol{z}_t)\right]}{\partial f_{\langle j-1 \rangle}^{\langle \alpha_q \rangle}(\boldsymbol{z}_t)} \Bigg|_{f_{\langle j-1 \rangle}^{\langle \alpha_q \rangle}(\boldsymbol{z}_t) = \hat{f}_{\langle j-1 \rangle}^{\langle \alpha_q \rangle}(\boldsymbol{z}_t)} \right\}. \tag{17}$$

The weak learner to make predictions is a regression tree fitted on a random subsample extracted from the original data, using the negative gradients as response variables and the predictors as input variables. More specifically, the predicted value $\hat{g}_t^{\langle \alpha_q \rangle}$ for the t^{th} negative gradient, given the predictors $\boldsymbol{z}_t$, can be written as:

$$\hat{g}_t^{\langle \alpha_q \rangle}(\boldsymbol{z}_t) = \sum_{s=1}^{S} \bar{g}^{\langle \alpha_q \rangle, \langle s \rangle} \cdot \mathrm{I}\left[\boldsymbol{z}_t \in \mathcal{R}_{\ell^{\langle s \rangle}}\right], \tag{18}$$

where $\bar{g}^{\langle \alpha_q \rangle, \langle s \rangle}$ is the average of the negative gradient values contained in the s^{th} leaf of the fitted tree, S is the total number of leaves, $\mathcal{R}_{\ell^{\langle s \rangle}}$ is the rectangular subspace domain corresponding to the s^{th} terminal leaf $\ell^{\langle s \rangle}$. The indicator function assumes value 1 if predictors $\boldsymbol{z}_t$ belong to the subspace $\mathcal{R}_{\ell^{\langle s \rangle}}$ (or, equivalently, if predictors $\boldsymbol{z}_t$ individuate the s^{th} leaf on the fitted tree), and 0 otherwise. At the j^{th} iteration, the updated weak learner is given by:

$$\hat{f}_{\langle j \rangle}^{\langle \alpha_q \rangle} = \hat{f}_{\langle j-1 \rangle}^{\langle \alpha_q \rangle} + \hat{\rho}_{\langle j \rangle}^{\langle \alpha_q \rangle}, \tag{19}$$

where $\hat{\rho}_{\langle j \rangle}^{\langle \alpha_q \rangle}$ is the gradient descent step size to update the estimate of $f^{\langle \alpha_q \rangle}$ at the j^{th} iteration. The gradient descent is obtained by adding the outcome of the regression tree, $\rho \cdot \hat{g}_t^{\langle \alpha_q \rangle}(\boldsymbol{z}_t)$, to the previous estimate $\hat{f}_{\langle j-1 \rangle}^{\langle \alpha_q \rangle}$, in order to get an improved estimate, i.e.:

$$\hat{\rho}_{\langle j \rangle}^{\langle \alpha_q \rangle} = \underset{\rho}{\operatorname{argmin}} \sum_{t \in \Omega^{\langle tr \rangle}} PS\left[P_t, \hat{f}_{\langle j-1 \rangle}^{\langle \alpha_q \rangle}(\boldsymbol{z}_t) + \rho \cdot \hat{g}_t^{\langle \alpha_q \rangle}(\boldsymbol{z}_t)\right]. \tag{20}$$

Since the value $\hat{g}_t^{\langle \alpha_q \rangle}(\boldsymbol{z}_t)$ is constant in the terminal leaf individuated by $\boldsymbol{z}_t$ (i.e., it is the average of the negative gradient values contained in the unique $\tilde{s}^{\text{th}}$ leaf, $\ell^{\langle \tilde{s}(\boldsymbol{z}_t) \rangle}$, individuated by $\boldsymbol{z}_t$), the problem in (20) can be solved separately for each s^{th} leaf subspace, yielding the simplified expression (Friedman, 2002, Buzna et al., 2020):

$$\hat{\rho}_{\langle j \rangle}^{\langle \alpha_q \rangle, \langle s \rangle} = \underset{\rho}{\operatorname{argmin}} \sum_{t \in \Omega^{\langle tr \rangle}} \left\{ PS\left[P_t, \hat{f}_{\langle j-1 \rangle}^{\langle \alpha_q \rangle}(\boldsymbol{z}_t) + \rho\right] \cdot \mathrm{I}\left[\boldsymbol{z}_t \in \mathcal{R}_{\ell^{\langle s \rangle}}\right] \right\}. \tag{21}$$

Hence, at the j^{th} iteration, the updated weak learner can be expressed as:

$$\hat{f}_{\langle j \rangle}^{\langle \alpha_q \rangle} = \hat{f}_{\langle j-1 \rangle}^{\langle \alpha_q \rangle} + \upsilon \cdot \sum_{s=1}^{S} \hat{\rho}_{\langle j \rangle}^{\langle \alpha_q \rangle, \langle s \rangle} \cdot \mathrm{I}\left[\boldsymbol{z}_t \in \mathcal{R}_{\ell^{\langle s \rangle}}\right], \tag{22}$$

where υ is the weight of each learner (called *shrinkage* or *leaning rate*), whose value is strictly related to the optimal number of iterations (R *gbm* package, 2020) since smaller values involve more iterations and usually more skilled forecasts (Persson et al., 2015).

An ending condition for the iterative procedure is reaching values of $\hat{\rho}_{\langle j \rangle}^{\langle \alpha_q \rangle, \langle s \rangle}$ smaller than a given threshold. Assuming that this is obtained at the iteration $\bar{\jmath}$, the prediction is:

$$\hat{P}_h^{\langle \alpha_q \rangle}(\boldsymbol{z}_h | \hat{f}_{\langle \bar{\jmath} \rangle}^{\langle \alpha_q \rangle}) = \hat{f}_{\langle \bar{\jmath} \rangle}^{\langle \alpha_q \rangle}(\boldsymbol{z}_h)\,, \tag{23}$$

and, since it is easy to verify that $\hat{f}_{\langle \bar{\jmath} \rangle}^{\langle \alpha_q \rangle}$ is estimated upon training data $\boldsymbol{P}^{\langle tr \rangle}, \boldsymbol{Z}^{\langle tr \rangle}$, the GBRT predictive α_q-quantile $\hat{P}_h^{\langle \alpha_q \rangle}$ of PV power for the time horizon h, it consequently depends on $\boldsymbol{P}^{\langle tr \rangle}$ and $\boldsymbol{Z}^{\langle tr \rangle}$, too:

$$\hat{P}_h^{\langle \alpha_q \rangle}\left[\boldsymbol{z}_h \mid \hat{f}_{\langle \bar{\jmath} \rangle}^{\langle \alpha_q \rangle}\left(\boldsymbol{P}^{\langle tr \rangle}, \boldsymbol{Z}^{\langle tr \rangle}\right)\right] = \hat{f}_{\langle \bar{\jmath} \rangle}^{\langle \alpha_q \rangle}(\boldsymbol{z}_h). \tag{24}$$

Note that the application of the gradient descent to search for the PS minimum (15) may lead to a local minimum. This notable drawback of the GBRT is typically solved through stochastic gradient descend or by bootstrapping. Bayesian bootstrap intrinsically mitigates this problem, as several replicates of the predictive α_q-quantile $P_h^{\langle \alpha_q \rangle}$ of PV power will be considered to build the final forecast, as it will be described in Section 3.4.

3.3. Quantile Regression Neural Network

QRNN exploits a neural network to generate predictive quantiles of PV power. It estimates conditional quantiles for specified values of quantile probability using regression equations and reproducing the behavior of the human brain to discern among the informative inputs and to produce an output. An efficient approach is the Monotone Composite Quantile Regression Neural Network (MCQRNN) proposed in (Cannon, 2018) which is based on the Multi-Layer Perceptron (MLP) neural network with partial monotonicity. It assumes the predictive α_q-quantile $\hat{P}_h^{\langle \alpha_q \rangle}$ of PV power coming from a weighted combination of L hidden layer outputs:

$$P_h^{\langle \alpha_q \rangle}(\boldsymbol{z}_h | \boldsymbol{\vartheta}^{\langle \alpha_q \rangle}) = \sum_{l=1}^{L}\left[\Phi\left(\sum_{c \in \Xi_1} z_{h,c} \cdot e^{\gamma_{c,l}^{\langle \alpha_q \rangle}} + \sum_{c \in \Xi_2} z_{h,c} \cdot \gamma_{c,l}^{\langle \alpha_q \rangle} + \tau_l^{\langle \alpha_q \rangle}\right) \cdot e^{\mu_l^{\langle \alpha_q \rangle}}\right] + \varepsilon^{\langle \alpha_q \rangle}, \tag{25}$$

where $\Phi(\cdot)$ is the function applied by each of the L neurons in the network hidden layer (in this paper, the hyperbolic tangent function), Ξ_1 is the set of indices for predictors monotonically increasing with the predictors, Ξ_2 is the corresponding set of indices for predictors. Ξ_2 does not have monotonicity constraints (note that $\boldsymbol{z}_h = \{z_{h,c}, c \in \Xi_1\} \cup \{z_{h,c}, c \in \Xi_2\}$), $\boldsymbol{\gamma}_1^{\langle \alpha_q \rangle}, \ldots, \boldsymbol{\gamma}_L^{\langle \alpha_q \rangle}$, $\boldsymbol{\mu}^{\langle \alpha_q \rangle} = \{\mu_1^{\langle \alpha_q \rangle}, \ldots, \mu_L^{\langle \alpha_q \rangle}\}$, $\boldsymbol{\tau}^{\langle \alpha_q \rangle} = \{\tau_1^{\langle \alpha_q \rangle}, \ldots, \tau_L^{\langle \alpha_q \rangle}\}$, and $\varepsilon^{\langle \alpha_q \rangle}$ are the parameters (all included in the vector $\boldsymbol{\vartheta}^{\langle \alpha_q \rangle}$, for clarity of representation) of the QRNN.

These parameters are once again estimated by minimizing the PS over a training dataset [27], i.e., by solving the following constrained optimization problem:

$$\widehat{\boldsymbol{\vartheta}}^{\langle\alpha_q\rangle} = \underset{\boldsymbol{\vartheta}^{\langle\alpha_q\rangle}}{\operatorname{argmin}} \sum_{t\in\Omega^{\langle tr\rangle}} PS\left[P_t, P_t^{\langle\alpha_q\rangle}\right],$$
$$\text{s.t.} \quad \frac{\partial P_t^{\langle\alpha_q\rangle}}{\partial z_{t,c}} \geq 0, \quad \forall c \in \Xi_1. \tag{26}$$

It is easy to verify that parameters $\widehat{\boldsymbol{\vartheta}}^{\langle\alpha_q\rangle}$ are estimated upon training data $\boldsymbol{P}^{\langle tr\rangle}, \boldsymbol{Z}^{\langle tr\rangle}$. The QRNN predictive α_q-quantile $\hat{P}_h^{\langle\alpha_q\rangle}$ of PV power for the time horizon h consequently depends on $\boldsymbol{P}^{\langle tr\rangle}$ and $\boldsymbol{Z}^{\langle tr\rangle}$, too:

$$\hat{P}_h^{\langle\alpha_q\rangle}\left[\boldsymbol{z}_h \mid \widehat{\boldsymbol{\vartheta}}^{\langle\alpha_q\rangle}\left(\boldsymbol{P}^{\langle tr\rangle}, \boldsymbol{Z}^{\langle tr\rangle}\right)\right] = \sum_{l=1}^{L}\left[\Phi\left(\sum_{c\in\Xi_1} z_{h,c}\cdot e^{\hat{\gamma}_{c,l}^{\langle\alpha_q\rangle}} + \sum_{c\in\Xi_2} z_{h,c}\cdot\hat{\gamma}_{c,l}^{\langle\alpha_q\rangle} + \hat{\tau}_l^{\langle\alpha_q\rangle}\right)\cdot e^{\hat{\mu}_l^{\langle\alpha_q\rangle}}\right] + \hat{\varepsilon}^{\langle\alpha_q\rangle}. \tag{27}$$

3.4. The role of the Bayesian bootstrap in the probabilistic forecasting system

As discussed above, in either the LQR, the GBRT or the QRNN the α_q-quantile $P_h^{\langle\alpha_q\rangle}$ of PV power for the target horizon h can be viewed as a function of the predictors $\boldsymbol{z}$ and of the PV power $\boldsymbol{P}$. In this sense, the predictive α_q-quantile $\hat{P}_h^{\langle\alpha_q\rangle}$ of PV power for the target horizon, h can be viewed as a function of the training data $\boldsymbol{Y}^{\langle tr\rangle} = [\boldsymbol{P}^{\langle tr\rangle}\ \boldsymbol{Z}^{\langle tr\rangle}]$, and thus it can be viewed as a target statistic calculated on M_{tr} variables, with occurrences contained in $\boldsymbol{Y}^{\langle tr\rangle}$. Therefore, the Bayesian bootstrap presented in Section 2 can be directly applied to evaluate the sample Bayesian bootstrap distribution of the predictive α_q-quantile $\hat{P}_h^{\langle\alpha_q\rangle}$ of PV power, with the following correspondences:

$$\begin{aligned}
&\boldsymbol{x} = \boldsymbol{y} = [\boldsymbol{P}\ \boldsymbol{z}], && \boldsymbol{X} = \boldsymbol{Y}^{\langle tr\rangle} = [\boldsymbol{P}^{\langle tr\rangle}\ \boldsymbol{Z}^{\langle tr\rangle}],\\
&\varphi(\boldsymbol{x}) = P_h^{\langle\alpha_q\rangle}(\boldsymbol{y}), && \hat{\varphi}(\boldsymbol{x}) = \hat{P}_h^{\langle\alpha_q\rangle}\left(\boldsymbol{Y}^{\langle tr\rangle}\right),\\
&M = M_{tr}, && N_b = N + 1.
\end{aligned} \tag{28}$$

The sample Bayesian bootstrap distribution is constituted by R replicates of the predictive α_q-quantile $P_h^{\langle\alpha_q\rangle}$ of PV power, i.e., $\widehat{\boldsymbol{P}}_h^{\langle\alpha_q\rangle} = \{\hat{P}_h^{\langle\alpha_q,1\rangle}, \dots, \hat{P}_h^{\langle\alpha_q,R\rangle}\}$. The necessary steps to calculate them are hereby summarized:

1) R weight samples $\boldsymbol{w}^{\langle 1\rangle}, \dots, \boldsymbol{w}^{\langle R\rangle}$ are independently drawn from distribution (7);

2) $F^{\langle 1\rangle}(\boldsymbol{x}), \dots, F^{\langle R\rangle}(\boldsymbol{x})$ are calculated by applying (2);

3) $G\left[F^{\langle 1\rangle}(\boldsymbol{x})\right], \dots, G[F^{\langle R\rangle}(\boldsymbol{x})]$ are calculated by applying either (11)-(13) for the LQR, (20)-(24) for the GBRT, or (26)-(27) for the QRNN;

4) $\hat{P}_h^{\langle\alpha_q,r\rangle} = G\left[F^{\langle r\rangle}(\boldsymbol{x})\right]$ for $r = 1, \dots, R$.

It is worth noting that the application of the Bayesian bootstrap to the predictive α_q-quantile $\hat{P}_h^{\langle\alpha_q\rangle}$ of PV power is a specific contribution of this research, since bootstrapping techniques are typically applied to the parameters of the underlying models rather than on the target variable. With the proposed approach there is no need to pass through the sample bootstrap distributions

of the parameters since the sample Bayesian bootstrap distribution of the predictive α_q-quantile $\hat{P}_h^{\langle \alpha_q \rangle}$ of PV power is directly provided, thus reducing the overall computational effort which is particularly important in short-term forecasting.

3.5. Optimizing the Bayesian-bootstrap-based predictions

Dealing with a sample bootstrap distribution of predictive quantiles may be not friendly for system operators, who are the recipients of the PV power forecasts and are usually unaware of the statistical background behind the predictions. Also, most of the probabilistic decision-making tools in power systems accept input probabilistic forecasts of PV power given either in terms of predictive distribution or in terms of a set of predictive quantiles (Lauret et al, 2019, Van der Meer et al., 2018). For this reason, two procedures to extract an optimal predictive quantile from the sample Bayesian bootstrap distribution are presented in this paper, in order to put the forecasting system in line with the needs of operators and practitioners.

The first procedure, called Sample Mean Bayesian Bootstrap (SM-BB), is quite naïve and it simply consists of picking the sample mean from the sample Bayesian bootstrap distribution $\widehat{\boldsymbol{P}}_h^{\langle \alpha_q \rangle}$ as the optimal predictive α_q-quantile of PV power $\hat{P}_h'^{\langle \alpha_q \rangle}$ for the target horizon h, i.e.:

$$\hat{P}_h'^{\langle \alpha_q \rangle} = \frac{1}{R} \sum_{r=1}^{R} \hat{P}_h^{\langle \alpha_q, r \rangle}. \tag{29}$$

The sample mean performs well in most scenarios; moreover, it does not exactly require a rigorous "optimization.

The second procedure, called Optimal Quantile Bayesian Bootstrap (OQ-BB), consists of picking a sample quantile from the sample Bayesian bootstrap distribution $\widehat{\boldsymbol{P}}_h^{\langle \alpha_q \rangle}$ as the optimal predictive α_q-quantile of PV power $\hat{P}_h'^{\langle \alpha_q \rangle}$ for the target horizon h, i.e.:

$$\hat{P}_h'^{\langle \alpha_q \rangle} = \hat{P}_h^{\langle \alpha_q, r^* \rangle}, \tag{30}$$

where $\hat{P}_h^{\langle \alpha_q, r^* \rangle}$ is the value that is smaller than a σ^* fraction of the samples in $\widehat{\boldsymbol{P}}_h^{\langle \alpha_q \rangle}$ or, equivalently, $100 \cdot (1 - \sigma^*)\%$ of the samples in $\widehat{\boldsymbol{P}}_h^{\langle \alpha_q \rangle}$ are greater than $\hat{P}_h^{\langle \alpha_q, r^* \rangle}$. The fraction σ^* is the result of an optimization problem that minimizes the PS over a validation dataset with indices $\Omega^{\langle va \rangle}$ (this validation dataset may or may not have overlaps with the training dataset; the latter option is preferable). It is:

$$\sigma^* = \underset{\sigma}{\operatorname{argmin}} \sum_{t \in \Omega^{\langle va \rangle}} \left\{ \alpha_q - \mathrm{I}\left[P_t \leq \hat{P}_t^{\langle \alpha_q, r^* \rangle} \right] \right\} \cdot \left(P_t - \hat{P}_t^{\langle \alpha_q, r^* \rangle} \right), \tag{31}$$

with $\hat{P}_t^{\langle \alpha_q, r^* \rangle} = \inf\{ \hat{P}_t^{\langle \alpha_q, r \rangle} \in \widehat{\boldsymbol{P}}_t^{\langle \alpha_q \rangle} : \hat{F}_{BB,t}^{\langle \alpha_q \rangle} \left(P_t \leq \hat{P}_t^{\langle \alpha_q, r \rangle} \right) \geq \sigma \}$, and $\hat{F}_{BB,t}^{\langle \alpha_q \rangle}$ is the cumulative distribution obtained from the sample Bayesian bootstrap distribution of the predictive α_q-quantile of PV power at time t.

3.6. Hints on the selection of the size of the sample Bayesian bootstrap distribution

The size R of the sample bootstrap distribution has an impact on the overall performance of the forecasting system. This topic has been discussed extensively in the literature on the traditional bootstrap framework, but there is no general agreement

about how the sample size should be arranged with respect to the number of available data M. Optimizing R through a random search upon a validation set is, in general, a good practice and this should also be valid for the Bayesian bootstrap; theoretically, there are no boundaries in which the optimal R should be searched. However, there are some practical limitations:

1) the Bayesian bootstrap is originally applied in this paper to a particular statistic, i.e., the predictive quantile of PV power, and therefore the function $G[\cdot]$ intrinsically contains the formulation of the training procedure of the probabilistic forecasting model. For models that require a non-trivial solution of the training procedure (as in the case of the GBRT and QRNN), increasing R determines an increased computational complexity that is not in line with some short-term forecasting lead times using standard workstations;

2) increasing R does not necessarily increase the performance of the forecasts. We found in our numerical experiments that optimal values for R are across a 1:100 ratio between R and M, as performance deteriorates with greater R.

For these reasons, the search for the optimal R is performed within this range.

4. Background of the performance assessment

The performance of Bayesian bootstrap in probabilistic PV power forecasting is assessed in a wide comparative framework. Several benchmarks and error indices and scores, presented in this Section, are exploited for this assessment.

4.1. Benchmarks

Several benchmarks are discussed in this paper to compare the outcomes of the Bayesian-bootstrap-based forecasts and to highlight the pros and cons with respect to existing literature.

The first group of benchmarks aims at evaluating how the Bayesian bootstrap performs with reference to the traditional bootstrap. This group, therefore, includes three forecasting systems (LQR-TB, GBRT-TB, and QRNN-TB) that apply the traditional bootstrap to build the sample traditional bootstrap distribution of the predictive quantiles of PV power, respectively applying an LQR, a GBRT and a QRNN model. The extraction of the optimal prediction from the traditional bootstrap distribution is performed applying the SM and the OQ procedures, as described in Section 3.5 for the Bayesian bootstrap. Therefore, the only difference with the presented Bayesian-bootstrap forecasts consists of the different bootstrap procedure applied in the first place.

The second group of benchmarks aims at evaluating if, in general, the bootstrap increases the performance or not. This group, therefore, includes three forecasting systems (LQR-NB, GBRT-NB, and QRNN-NB) that directly predict the quantiles of PV power, respectively applying an LQR, a GBRT and a QRNN model, without any bootstrap.

The third group of benchmarks is instead based on persistence models, and they are provided as an unbiased reference for the performance evaluation. This group includes two benchmarks: the PM1 that assumes the predictive quantiles for the target horizon equal to the last observed PV power, i.e.:

$$\hat{P}_h^{\langle \alpha_q \rangle} = P_{h-k}, \qquad \forall q = 1, \dots, Q, \tag{32}$$

and the PM2 that assumes the predictive quantiles for the target horizon equal to the PV power observed in the same time slot of the day before, as in the smart persistence framework. For an hourly time resolution, e.g., the PM2 returns:

$$\hat{P}_h^{\langle \alpha_q \rangle} = P_{h-24}, \qquad \forall q = 1, \dots, Q. \tag{33}$$

4.2. Probabilistic error indices and scores

In this paper, three error indices are used to compare the accuracy of the proposed forecasting method with the other methods which have been used as benchmarks. In what follows, the definition of the PS metric is first recalled (Lauret et al., 2019). Then, the Average Absolute Coverage Error (AACE) and the Prediction Intervals Normalized Width (PINAW) are briefly introduced.

A. Pinball score

PS allows addressing the accuracy of the prediction by evaluating, at the same time, the reliability and the sharpness of the forecasted values (Bracale et al., 2016, Gneiting and Raftery, 2007). It is used in all the three considered models as the loss function to be minimized to train the corresponding parameters as it is a negatively oriented error measure (i.e., a smaller PS indicates a better forecast performance). It is here recalled that PS is defined as:

$$PS\left[P_h, \hat{P}_h^{\langle \alpha_q \rangle}\right] = \left\{\alpha_q - \mathrm{I}\left[P_h \leq \hat{P}_h^{\langle \alpha_q \rangle}\right]\right\} \cdot \left(P_h - \hat{P}_h^{\langle \alpha_q \rangle}\right). \tag{34}$$

To obtain a measure of the forecast performance in a comprehensive manner, the value of PS can be evaluated by averaging the values it assumes across multiple forecast issues and summing over the Q quantiles. In the numerical experiments, a normalized version of the PS is used for evaluating performance in the test period. The Normalized Pinball Score (NPS) is:

$$NPS\left[P_h, \hat{P}_h^{\langle \alpha_q \rangle}\right] = \frac{PS\left[P_h, \hat{P}_h^{\langle \alpha_q \rangle}\right]}{\bar{P}_{\text{rated}}}, \tag{35}$$

where $\bar{P}_{\text{rated}}$ is the rated power of the PV system.

B. Average Absolute Coverage Error

The AACE is used to assess the reliability of the forecasting method, by quantifying the difference of the predicted values and the nominal coverages of the predictive quantiles (Alfieri and De Falco, 2020). AACE can only be formulated for multiple forecast issues. For a test set with indices $\Omega^{\langle te \rangle}$, the estimated α_q-coverage $\hat{\alpha}_q$ is provided by:

$$\hat{\alpha}_q = \frac{1}{M_{te}} \sum_{t \in \Omega^{\langle te \rangle}} \mathrm{I}\left[P_t, \hat{P}_t^{\langle \alpha_q \rangle}\right], \tag{36}$$

with M_{te} the size of the considered test set. The absolute coverage error on the nominal α_q-quantile, $ACE^{\langle \alpha_q \rangle}$, is defined as:

$$ACE^{\langle \alpha_q \rangle} = \left|\alpha_q - \hat{\alpha}_q\right|. \tag{37}$$

and the percentage value of AACE, $AACE_{\%}$, across the Q coverages can be easily derived as a percentage value of $ACE^{\langle \alpha_q \rangle}$, as:

$$AACE_{\%} = \frac{100}{Q} \cdot \sum_{q=1}^{Q} ACE^{\langle \alpha_q \rangle}. \quad (38)$$

AACE is a negatively oriented metric, i.e., smaller is the value it assumes, more reliable is the forecast method.

C. Prediction intervals normalized width

The PINAW is used to assess the sharpness of the forecasting method, by quantifying the width of the prediction intervals (Alfieri and De Falco, 2020). It is a property of the forecast by itself, so this index is not calculated considering the actual PV power outcomes. For a test set of size M_{te} with indices $\Omega^{\langle te \rangle}$, the PINAW at the nominal prediction interval rate λ is:

$$PINAW^{\lambda} = \frac{1}{M_{te}} \sum_{t \in \Omega^{\langle te \rangle}} \frac{\hat{P}_t^{\langle 0.5+\lambda/2 \rangle} - \hat{P}_t^{\langle 0.5-\lambda/2 \rangle}}{\bar{P}_{\text{rated}}}, \quad (39)$$

PINAW is a negatively oriented metric, i.e., smaller is the value it assumes, sharper are the forecasts.

5. Numerical experiments

Bayesian bootstrap in probabilistic PV power forecasting is evaluated on an actual PV dataset collected at HEIG-VD, Switzerland (ReIne Lab (Carpita et al., 2019) from 12 roof-top PV panels with a-Si/μC-Si dual junction technology. For each panel, the stable peak power, at maximum power point (MPP) is $P_{MPP} = 100\ W$. The PV panels are connected to the AC grid through a 1.3 kW AC/DC inverter. The data set includes both the electrical system and meteorological data collected at every one minute. The electrical system data set composed of the measurements of voltage, current and power on DC side of the inverter (i.e., output of the PV panels). The meteorological data set composed of all the relevant weather measurements such as global, direct and diffuse irradiations, wind speed and wind direction, wind speed, external temperature, humidity, pressure, etc. Figure 1 shows the PV panels as well as some of weather measurement devices.

Fig. 1 – PV system and measurement setup at HEIG-VD, Switzerland.

The NWPs used in the experiments are taken from the European Centre for Medium-range Weather Forecast (ECMWF) (ECMWF website, 2020). All the NWPs belong to the midnight run, i.e., they are issued at midnight and cover the 24 hours of the following day.

All the data are averaged to obtain an hourly time resolution. They are normalized to their respective minimum and maximum values to be processed by the forecasting models.

Data are stored from February 1, 2016 to November 30, 2018, for a total number of 24816 occurrences. In these experiments, the training set covers until January 31, 2018 and it is $\Omega^{\langle tr \rangle} = \{t: 1 \leq t \leq 17544\}$, whereas the test set covers the remaining months in 2018 and is $\Omega^{\langle te \rangle} = \{t: 17545 \leq t \leq 24816\}$. The validation set $\Omega^{\langle va \rangle}$ used in the OQ-BB procedure to optimize the extraction of the final prediction from the sample Bayesian bootstrap distribution is applied on a rolling monthly window: for example, the validation set $\Omega^{\langle va \rangle} = \{t: 16801 \leq t \leq 17544\}$ that corresponds to data in January 2018 is used to optimize the extraction of the final prediction for February 2018, the validation set $\Omega^{\langle va \rangle} = \{t: 17545 \leq t \leq 18264\}$ that corresponds to data in February 2018 is used to optimize the extraction of the final prediction for March 2018, and so on.

The 1-hour-ahead probabilistic forecasts are generated by $Q = 19$ predictive quantiles at nominal coverages $\alpha_1, \dots, \alpha_{19} = 0.05, 0.10, \dots, 0.90, 0.95$. For the considerations in Section 3.6, the sample size of the Bayesian bootstrap distribution is $R = 100$ and is kept at this value for all the experiments. All forecasts are generated using an i7-6700HQ CPU @2.60GHz equipped with 16 GB RAM in R, with the packages *bayesboot* (R *bayesboot* package, 2018), *quantreg* (R *quantreg* package, 2020), *qrnn* (R *qrnn* package, 2019) and *gbm* (R *gbm* package, 2020). In any case, the time required to generate forecasts was in line with the requirements driven by the 1-hour lead time.

Table I shows the NPS, the AACE and the PINAW obtained using the Bayesian-bootstrap-based forecasting systems and the benchmarks, averaged across the test set. Bold values in Table I indicate the best performance for each model family.

The application of the Bayesian bootstrap to either LQR, GBRT or QRNN allows increasing the skill of the forecasts, as the NPS are the smallest in the three cases. In particular, the OQ-BB procedure always outperforms the SM-BB procedure, suggesting that the optimization of the quantile extraction is mandatory to pick the best from the Bayesian bootstrap samples.

Table I. Performance of the probabilistic forecasts averaged across the test set. Bold values indicate the best performance for each model family.

Model family	**Bootstrap**	**Forecasting system**	**Error score/index**			
			NPS [-]	AACE [%]	$PINAW_{10}$ [-]	$PINAW_{90}$ [-]
LQR	Bayesian	LQR-SM-BB	0.184	2.89	0.713	**11.637**
		LQR-OQ-BB	**0.183**	**1.42**	**0.632**	11.980
	Traditional	LQR-SM-TB	0.184	2.79	0.699	12.012
		LQR-OQ-TB	0.184	2.09	0.675	12.365
	None	LQR-NB	0.186	2.99	0.749	12.312
GBRT	Bayesian	GBRT-SM-BB	0.188	3.91	**0.560**	**7.938**
		GBRT-OQ-BB	**0.185**	**0.97**	0.675	11.481
	Traditional	GBRT-SM-TB	0.192	3.10	0.684	8.289
		GBRT-OQ-TB	0.190	1.23	0.731	11.705
	None	GBRT-NB	0.194	4.11	0.830	12.172
QRNN	Bayesian	QRNN-SM-BB	0.188	3.64	0.797	**7.135**
		QRNN-OQ-BB	**0.186**	**2.02**	**0.716**	10.200
	Traditional	QRNN-SM-TB	0.187	3.89	0.831	7.427
		QRNN-OQ-TB	**0.186**	2.08	0.782	8.985
	None	QRNN-NB	0.189	3.68	0.827	7.461
PM1			0.522	-	-	-
PM2			0.595	-	-	-

The NPS improvements brought by the Bayesian bootstrap, with respect to the traditional bootstrap, are about 0.5%, and 2.7% on the LQR and GBRT, respectively; there is no improvement instead on the QRNN. Compared to the rough application of the underlying probabilistic models, the Bayesian bootstrap allows reducing the NPS by about 1.1%, 4.7% and 1.6%. The NPS of the OQ-BB procedure is roughly one third of the NPS of the PM1 and PM2 benchmarks.

It is also important to observe that the Bayesian bootstrap returns the most reliable forecasts, as the LQR-OQ-BB, GBRT-OQ-BB, and QRNN-OQ-BB forecasting systems generate the smallest AACE index, which notably drops below 1% for the GBRT case. The LQR-OQ-TB, GBRT-OQ-TB, and QRNN-OQ-TB forecasting systems show also small AACE indices, denoting once again that the OQ procedure is significant for the performance of the predictions.

With reference to the sharpness, the Bayesian bootstrap returns also the forecasts with the smallest PINAW indices, although this is mostly achieved by the SM-BB procedure which, in turn, is less reliable. This, however, is not surprising, as reliability and sharpness are typically in contrast to each other [29]. Nevertheless, the PINAW of the LQR-OQ-BB, GBRT-OQ-BB, and QRNN-OQ-BB forecasting systems are in line with those of the benchmarks, suggesting that the increased reliability does not make the forecast unnecessarily less sharp.

The reliability diagrams are shown for the LQR-OQ-BB, the GBRT-OQ-BB and the QRNN-OQ-BB in Figure 2(a), 2(b) and 2(c), respectively. The graphical inspection of the reliability diagrams denotes that forecasts are calibrated, as the estimated coverages tend to lie along the bisector curve. GBRT-OQ-BB forecasts only marginally deviate from the ideality, as confirmed by the smallest AACE index (0.97%) reported in Table I.

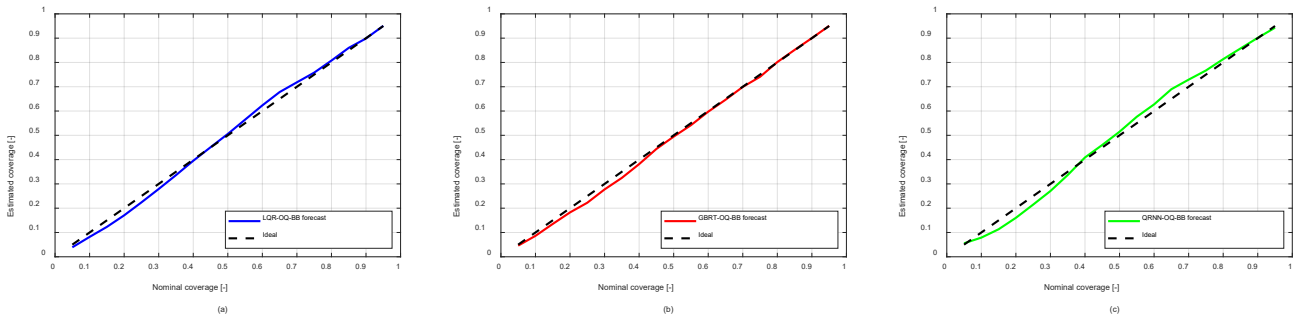
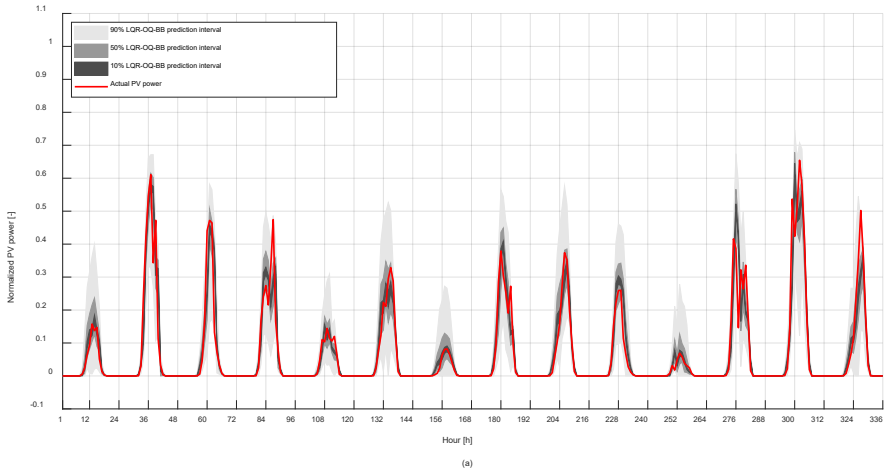
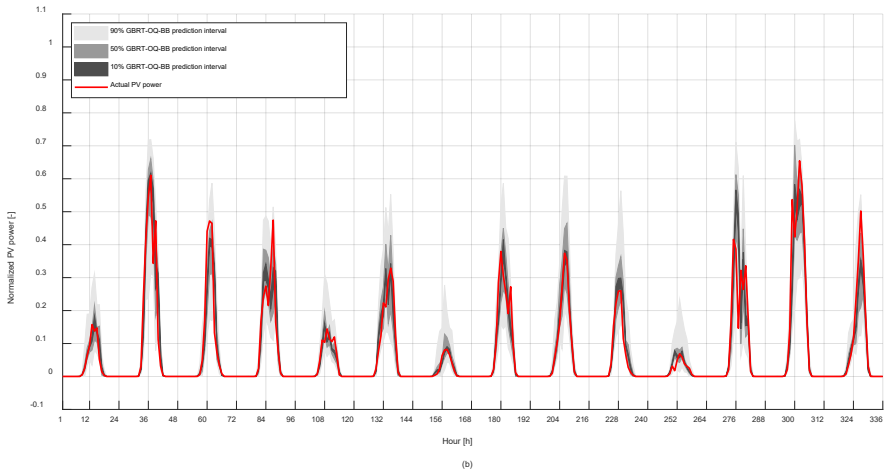


Fig. 2 – Reliability diagrams of the LQR-OQ-BB forecasts (a), GBRT-OQ-BB forecasts (b), and QRNN-OQ-BB forecasts (c).

LQR-OQ-BB forecasts, GBRT-OQ-BB forecasts and QRNN-OQ-BB forecasts during the first two weeks of the test period are plotted versus time and compared to the actual PV power in Figure 3(a), 3(b) and 3(c), respectively. Figures show similar behaviors of the forecast versus time and demonstrate the effectiveness of the proposed approach with values of the prediction intervals that, in most cases, include the actual PV power.





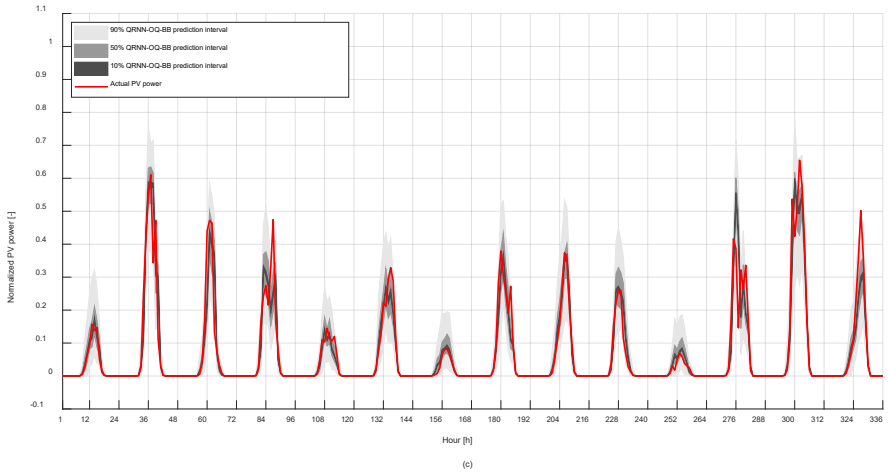


Fig. 3 – LQR-OQ-BB forecasts (a), GBRT-OQ-BB forecasts (b), and QRNN-OQ-BB forecasts (c) during the first two weeks of the test period.

6. Conclusions

This paper addresses Bayesian bootstrap in real-time probabilistic PV power forecasting. Three forecasting systems based on Bayesian bootstrap and underlying probabilistic models (LQR, GBRT and QRNN) are developed and compared with their non-bootstrapped counterparts and with their traditional-bootstrapped counterparts, to evaluate the role of Bayesian bootstrap

in increasing the skill of predictions. The Bayesian bootstrap is arranged in the forecasting systems in order to directly address the predictive PV power quantiles rather than the model parameters, thus allowing for a direct estimation of the target variable. A dedicated procedure for the extraction of the optimal sample quantile from the sample Bayesian bootstrap distribution is also prepared, and it proves to be significant in the overall improvement of the forecast skill.

Several benchmarks and error indices are used to assess the Bayesian bootstrap in probabilistic PV power forecasting. The systems are tested on an actual PV dataset. The results of the numerical experiments show that the Bayesian bootstrap is effective in adding skill to the final predictions, whatever the underlying probabilistic forecasting model is considered. Combined with the OQ procedure for the optimal quantile extraction, the Bayesian bootstrap reduces the NPS up to 2.7% with respect to the traditional bootstrap and up to 4.7% with respect to the rough application of the underlying probabilistic models, with excellent reliability and good sharpness of the probabilistic predictions.

Future research may address the inclusion of Bayesian bootstrap in other probabilistic energy forecasting systems or the development of new frameworks for the Bayesian bootstrap formulation.

Acknowledgment

This research was supported through the ERA-NET Smart Energy Systems Regsys joint program 2019 in the frame of the project “DiGRiFlex-Real time Distribution GRid control and Flexibility provision under uncertainties.”

Declaration of competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alexey, N., Alois, K., 2013. Gradient boosting machines, a tutorial. Frontiers Neurorobot. 7.

Alfieri, L., De Falco, P., 2020. Wavelet-based decompositions in probabilistic load forecasting. IEEE Trans. Smart Grid 11(2), 1367-1376.

Almeida, M.P., Perpiñán, O., Narvarte, L., 2015. PV power forecast using a nonparametric PV model. Solar Energy 115, 354-368.

Bessa, R.J., et al., 2017. Towards improved understanding of the applicability of uncertainty forecasts in the electric power industry. Energies 10, 1402.

Bozorg, M., et al., 2020. Bayesian bootstrap quantile regression for probabilistic photovoltaic power forecasting. J. Protect. Control Modern Power Syst. 5(1), 1-12.

Bracale, A., Carpinelli, G., De Falco, P., 2016. A probabilistic competitive ensemble method for short-term photovoltaic power forecasting. IEEE Trans. Sustain. Energy 8(2), 551-560.

Buzna, L., et al., 2020. Hierarchical probabilistic electric vehicle load forecasting. Submitted to Appl. Energy.

Cannon, A.J., 2018. Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes. Stoch. Environm. Res. Risk Assess. 32(11), 3207-3225.

Carpita, M., et al., 2019. ReIne, a flexible laboratory for emulating and testing the Distribution grid. 21st Europ. Conf. Power Electron. Appl. (EPE'19 ECCE Europe), Genova, Italy, 1-6.

Chu, Y., Li, M., Pedro, H.T.C., Coimbra, C.F.M., 2015. Real-time prediction intervals for intra-hour DNI forecasts, Renew. Energy 83, 234-244.

Clyde, M.A., and Lee, H.K.H., 2001. Bagging and the Bayesian bootstrap. Artificial Intelligence and Statistics, eds. Richardson, T. & Jaakkola, T. New York: Elsevier, 169-174.

Doubleday, K., Jascourt, S., Kleiber, W., Hodge, B., 2020. Probabilistic solar power forecasting using Bayesian model averaging. IEEE Trans. Sustain. Energy, in press.

ECMWF website, 2020. European Centre for Medium-range Weather Forecasts. Available online: https://www.ecmwf.int/ (Accessed on October 1 2020).

Fernandez-Jimenez, L.A., et al., 2017. Day-ahead probabilistic photovoltaic power forecasting models based on quantile regression neural networks. 2017 Europ. Conf. Electr. Eng. Comp. Sci. (EECS 2017), Bern, Switzerland, 89-294.

Friedman, J., 2002. Stochastic gradient boosting. Comput. Statist. Data Analysis 38(4), 367-378.

Gneiting, T, Raftery, A.E., 2007. Strictly proper scoring rules, prediction, and estimation. J. Am. Statist. Assoc. 102(477), 359-378.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The elements of statistical learning: Data mining, inference, and prediction, 2nd edition. Springer Series in Statistics.

Hong, T., et al., 2016. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. Int. J. Forecast 32(3), 896-913.

Juban, R., Ohlsson, H., Maasoumy, M., Poirier, L., Zico Kolter, J., 2016. A multiple quantile regression approach to the wind, solar, and price tracks of GEFCom2014. Int. J. Forecast. 32(3), 1094-1102.

La Salle, J.L.G., Badosa, J., David, M., Pinson, P., Lauret, P., 2020. Added-value of ensemble prediction system on the quality of solar irradiance probabilistic forecasts. Renew. Energy, in press.

Lauret, P., David, M., Pedro, H.T.C., 2017. Probabilistic solar forecasting using quantile regression models Energies 10, 1591.

Lauret, P., David, M., Pinson, P., 2019. Verification of solar irradiance probabilistic forecasts. Solar Energy 194, 254-271.

Lee, H., Lee, B., 2019. Confidence-aware deep learning forecasting system for daily solar irradiance. IET Renew. Power Gen. 13(10), 1681-1689.

Pedro, H.T.C., Coimbra, C.F.M., David, M., Lauret, P., 2018. Assessment of machine learning techniques for deterministic and probabilistic intra-hour solar forecasts. Renew. Energy 123, 191-203.

Persson, C., Bacher, P., Shiga, T., Madsen, H., 2015. Multi-site solar power forecasting using gradient boosted regression trees. Solar Energy 150, 423-436.

R *bayesboot* package, 2018. An implementation of Rubin's (1981) Bayesian bootstrap. Available online: https://CRAN.R-project.org/package=bayesboot (Accessed on October 1 2020).

R *gbm* package, 2020. Generalized boosted regression models. Available online: https://CRAN.R-project.org/package=gbm (Accessed on October 1 2020).

R *qrnn* package, 2019. Quantile regression neural network. Available online: https://CRAN.R-project.org/package=qrnn (Accessed on October 1 2020).

R *quantreg* package, 2020. Quantile regression. Available online: https://CRAN.R-project.org/package=quantreg (Accessed on October 1 2020).

Ren, Y., Suganthan, P.N., Srikanth, N., 2015. Ensemble methods for wind and solar power forecasting—A state-of-the-art review. Renew. Sustain. Energy Rev. 50, 82-91.

Rubin, D.B., 1981. The Bayesian bootstrap. Annals Statist. 9(1), 130-134.

Van der Meer, D.W., Widén, J., Munkhammar, J., 2018. Review on probabilistic forecasting of photovoltaic power production and electricity consumption. Renew. Sustain. Energy Rev. 81, 1484-1512.

Vergura, S., Vacca, F., 2009. Bootstrap technique for analyzing energy data from PV plant. 2009 Int. Conf. Clean Electr. Power (ICCEP 2009), Capri, Italy, 268-275.

Voyant, C., et al., 2017. Machine learning methods for solar radiation forecasting: A review. Renew. Energy 105, 569-582.

Voyant, C., et al., 2018. Prediction intervals for global solar irradiation forecasting using regression trees methods. Renew. Energy 126, 332-340.

Wen, T., et al., 2020. Performance evaluation of probabilistic methods based on bootstrap and quantile regression to quantify PV power point forecast uncertainty. IEEE Trans. Neural Networks Learn. Syst. 31(4), 1134-1144.