Applied Intelligence
https://doi.org/10.1007/s10489-022-03958-7

Deep neural networks for the quantile estimation of regional renewable energy production

Antonio Alcántara[1] · Inés M. Galván[2] · Ricardo Aler[2]

Accepted: 2 July 2022

Abstract

Wind and solar energy forecasting have become crucial for the inclusion of renewable energy in electrical power systems. Although most works have focused on point prediction, it is currently becoming important to also estimate the forecast uncertainty. With regard to forecasting methods, deep neural networks have shown good performance in many fields. However, the use of these networks for comparative studies of probabilistic forecasts of renewable energies, especially for regional forecasts, has not yet received much attention. The aim of this article is to study the performance of deep networks for estimating multiple conditional quantiles on regional renewable electricity production and compare them with widely used quantile regression methods such as the linear, support vector quantile regression, gradient boosting quantile regression, natural gradient boosting and quantile regression forest methods. A grid of numerical weather prediction variables covers the region of interest. These variables act as the predictors of the regional model. In addition to quantiles, prediction intervals are also constructed, and the models are evaluated using different metrics. These prediction intervals are further improved through an adapted conformalized quantile regression methodology. Overall, the results show that deep networks are the best performing method for both solar and wind energy regions, producing narrow prediction intervals with good coverage.

Keywords Deep neural networks · Prediction intervals · Probabilistic forecasting · Quantile estimation · Regional renewable energy forecasting

1 Introduction

In the last few years, there has been a large increase in the installed capacity of both wind and solar renewable energy. Wind and solar energy are nondispatchable energy sources, which means that they are not under the control of an operator; instead, these energy sources depend on weather conditions. This dependence makes the integration of wind and solar energy into the electricity grid more difficult than operable sources. Given that the amount of energy to be generated cannot be controlled, the only alternative is to forecast it with as much accuracy as possible. Numerical weather prediction (NWP) systems, which are based on mathematical/physical models of the atmosphere, are one of the most accurate ways to predict meteorological variables. However, in electricity generation, the most relevant dependent variable is how much electricity will be generated. One way of determining this is to couple NWP systems with machine learning models. The goal of the latter is to find the relation between NWP variables (the inputs to the model) and the electricity produced (the output, which is the dependent variable). An example of this approach can be found in [1].

However, most works deal with point or deterministic forecasts. It is currently becoming increasingly important to estimate the uncertainty associated with renewable energy forecasts [2]. Such forecasts, which are known as probabilistic forecasts, are more informative than the forecasts obtained from deterministic models. Several works have shown how probabilistic renewable energy forecasts allow

✉ Antonio Alcántara
antalcan@est-econ.uc3m.es

Inés M. Galván
igalvan@inf.uc3m.es

Ricardo Aler
aler@inf.uc3m.es

[1] Statistics Department, University Carlos III of Madrid, Av. de la Universidad 30, Leganés, 28911, Madrid, Spain

[2] Computer Science Department, University Carlos III of Madrid, Av. de la Universidad 30, Leganés, 28911, Madrid, Spain

Published online: 02 August 2022

for improvements in the management of power systems [3], the participation in the electricity market [4–6], and the bidding strategy of ancillary services of renewable power plants [7].

Probabilistic forecasts can be represented in different ways. Sets of quantiles are one of the most widely used representations of the predicted probability distribution [8]. A well-known method to estimate quantiles is to minimize the quantile loss using (linear) quantile regression, where linear models are trained for each of the quantiles to be estimated. It is important to remark that these are conditional quantiles (the model outputs the quantile, which is conditioned to the inputs/independent variables). However, in quantile regression, it is assumed that the relation between inputs and outputs is linear. For nonlinear relationships, other machine learning methods can be used. For instance, support vector machines can be extended to quantile regression by using quantile loss as a penalization term [9]. Random forests can be easily extended so that quantiles are output instead of deterministic predictions [10]. Gradient boosting techniques can be formulated as gradient descent optimization, and therefore, they can also return conditional quantiles by minimizing quantile loss [11]. A disadvantage of gradient boosting as well as linear quantile regression and support vector machines is that a different model has to be fit for each different quantile. The recently introduced natural gradient boosting method, which follows the general gradient boosting framework, can also be used to estimate quantiles; in this case, all quantiles can be provided using a single model [12]. All the nonlinear methods for quantile estimation described above have been used in recent energy forecasting work [13–16] for SVRQR, QRF, GBR, and NGB.

Deep neural networks (DNNs) are nonlinear models that have been very successful in recent years in many research fields, such as computer vision [17–20], natural language processing [21, 22] and renewable energy forecasting [23–26]. In [27], the most widely used methods in power research, such as convolutional neural networks (CNNs), autoencoders and deep belief networks, were reviewed. However, deep learning has been mainly used for point forecasting. For example, in [28], CNNs were employed for wind power point prediction. In [29], similar research was carried out; here, dense fully-connected neural networks were utilized to forecast wind power for a single wind farm. A hybrid LSTM-CNN method was employed in [30] to make point predictions of solar power, and LSTM models were also studied in [31] for short-term renewable electricity generation for a location. Apart from the most common renewable energy sources (i.e., solar and wind sources), the modeling of hydrogen production has also been considered using DNNs [32] but not from a probabilistic perspective.

Given that the most common training method of neural networks is gradient descent, these networks can also be used to obtain conditional quantiles by minimizing quantile loss. As a result, probabilistic predictions can be obtained. Despite their overall good performance, neural networks have not received much attention for comparative studies of probabilistic forecasting of renewable energies.

For instance, [33] made a comparison of several methods for computing probabilistic forecasts, but no neural networks were used. They started with several point forecast methods, including decision trees, nearest neighbors, gradient boosting, random forests, and lasso/ridge regression, and used some ensemble techniques to obtain quantiles for probabilistic solar energy forecasting. Additionally, in [34], different methods, such as decision trees, random forests and gradient boosting together with bootstrapping, were compared for the construction of probabilistic forecasts, but again, no neural networks were used in the study. In other studies [35, 36], in addition to random forests and gradient boosting decision trees, neural networks were used for quantile estimation. However, these neural architectures only contained one or two hidden layers [37, 38], and deep network performance was not studied.

In this article, we propose the use of deep neural networks (networks with more than 2 layers) for the quantile estimation of renewable energy (both solar and wind energy). Instead of estimating a single quantile as in other works [39, 40], the proposed quantile regression deep neural network (QRDNN) model has been designed to estimate multiple quantiles. In previous works, a different network was trained for each quantile to be estimated, which requires a large computational effort. The QRDNN model outputs all required quantiles using a single model, hence saving computational time. The combination of multiple layers and multiple output quantiles allows for complex nonlinear processing in the initial layers, while the last layer is used to adapt to each of the quantiles.

Pairs of quantiles can be used as the lower and upper limits of prediction intervals (PIs), which are widely used to represent the uncertainty of the dependent variable with a given probability. PIs should be as narrow as possible. However, this property is not directly considered when estimating quantiles. Therefore, PIs obtained from quantiles may be wider than necessary. In this work, QRDNN has been extended using conformalized quantile regression (CQR) [41], which allows the PIs obtained from the quantiles to be calibrated. The CQR is a very recently introduced calibration method that has seldom been used for deep networks [42]. Additionally, CQR has been adapted to the power generation problem addressed in this work by using several time prediction horizons. This is achieved by computing conformity scores that are dependent on the time horizon. This allows the separate adaptation of PIs to the characteristics of each time horizon.

The field of application for this work is probabilistic forecasting at the regional level. Most works deal with energy forecasting at the local level (e.g., a single wind farm or photovoltaic plant), but for some applications, electricity production is required to be aggregated at the regional level (e.g., areas, regions, or countries) [43–46]. In regional forecasting, geographical dispersion of plants in the region provides a balancing effect that results in a lower variability on the energy production, compared to the production of individual plants (solar or eolic). On the other hand, regional forecasting has some particular issues, such as maintenance operations, or down-regulation of individual plants, that add noise to the electricity production data. To empirically study regional forecasting, quantile models are obtained for the electricity production in four provinces in Spain at different forecasting horizons. To obtain a greater understanding of deep networks for renewable probabilistic forecasting, the two most important renewable energies, solar (Ciudad Real and Córdoba provinces) and wind (Granada and Lugo provinces) energies, are studied.

In summary, the main contributions of this article are:

- The combination of deep neural networks containing multiple layers and multiple quantiles at the output (QRDNN) are used to estimate a set of quantiles, which allows the estimation of a set of PIs.
- The conformalized quantile regression method is applied to calibrate multiple PIs obtained from the quantiles at the network output and its adaptation is applied to multiple time prediction horizons.
- An exhaustive comparative study in the context of regional renewable energy forecasting for both solar and wind energy is conducted. The performance of QRDNN has been compared with linear quantile regression (LQR) and state-of-the-art methods, such as support vector quantile regression (SVQR), gradient boosting quantile regression (GBQR), natural gradient boosting (NGB) and quantile regression forests (QRFs). Systematic hyperparameter tuning by a grid search is used for all methods. This comparison has been made using metrics related to quantile estimation as well as metrics related to the goodness of the PIs obtained from the quantiles.

The structure of this article is as follows. In Section 2, the meteorological and production datasets are described. In Section 3, the machine learning methods employed in the article are introduced. In Section 4, the methodology, models, metrics, and evaluation procedure are presented. In Section 5, the obtained results are documented and discussed. Finally, in Section 6, the main conclusions of this work are drawn.

2 Data

As previously mentioned, NWP variables (independent variables) are used as the inputs to predict the amount of renewable energy (solar or wind) generated in a region. Regarding the independent variables, an observational spatial grid is set across different Spanish regions ("provincias") from which we will be able to obtain these variables. This means that for every point on the observational grid, a complete set of NWP variables will be collected.

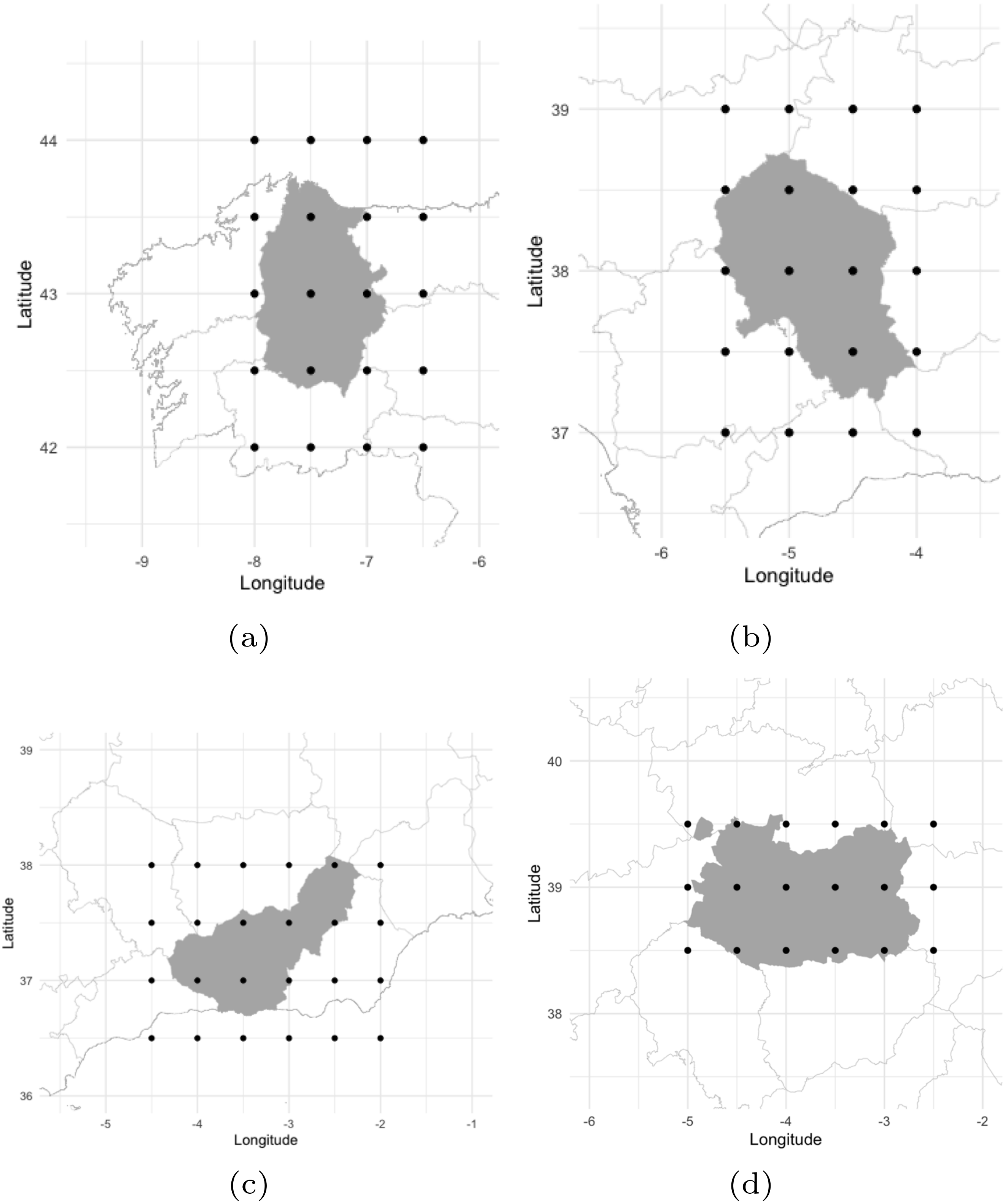
Data in the netCDF4 format are provided by the European Centre for Medium-Range Weather Forecasts (ECMWF) in the ERA5 database [47]. Overall, it is possible to obtain two data products: the ensemble mean and reanalysis data. The former represents the actual meteorological forecasts for each of the variables, which are provided as the mean of a forecast ensemble (data from the variables forecast by NWP at different time horizons and at several locations in the spatial grid). Additionally, reanalysis data are posterior calibrations produced with the aim of reducing forecasting errors.

While a spatial resolution of 0.25° × 0.25° is allowed for the reanalysis data, a resolution of 0.5° × 0.5° is provided for the ensemble mean data. Furthermore, reanalysis data are provided hourly, while the ensemble mean data are obtained every 3 hours beginning at 00:00 hours. However, in this article, some preliminary tests were made, suggesting that the uncertainty of the ensemble mean data allows for better modeling of the energy generation uncertainty. Therefore, the dataset has been constructed with the ensemble mean data. The NWP variables are extracted from a spatial grid with a 0.5° × 0.5° resolution.

We define four grid extensions to cover the majority of the four regions (Spanish provinces). The grids in the regions of Córdoba and Ciudad Real are employed for solar energy prediction. In addition, the grids in Lugo and Granada are used for wind energy prediction.

Figure 1 shows the observational grid for these four regions. The grid in Lugo includes longitudes from -8º to -6.5º and latitudes from -8º to 44º. In Córdoba, the grid includes longitudes from -5.5º to -4º and latitudes from 37º to 39º. In Granada, the grid spans LON from -4.5º to -2º and LAT from 36.5º to 38º. Finally, the grid in Ciudad Real includes LON from -5º to -2.5º and LAT from 38.5º to 39.5º.

Additionally, data for the dependent variable (generated energy) is obtained from the open data portal ESIOS of the Spanish regulator Red Eléctrica Española [48]. Within this portal, users can obtain data related to energy consumption, generation, and exchange, among other indicators. Electricity generation data are provided in hourly intervals. In addition, data can be filtered by the type of production (solar or wind in our design) and by region. Therefore, we



(a)

(b)

(c)

(d)

Fig. 1 Observational grids for (a) Lugo, (b) Córdoba, (c) Granada, (d) Ciudad Real

have selected the type of energy and the desired temporal set according to our selected regions.

We now explain how the complete dataset is built. The data provided by ECMWF must be transformed to obtain a 2-dimensional data matrix with observations in the rows and variables in the columns. NWP variables, which were provided by ECMWS in netCDF4 format, are contained in a three dimensional array. Each variable is measured at a specific latitude, longitude, and time. An arrangement is made so that every time point is an observation and every different variable X_i in each latitude j and longitude k is an input. For example, if we have N meteorological variables in a $j \times k$ spatial grid, the procedure will allow us to obtain a set of T observations (rows) and $N \times j \times k$ independent variables (columns).

Specifically, for our purpose, the variables shown in Table 1 are utilized. These selections are made according to other research in regional point energy prediction [1] that resulted in successful outcomes.

The ECMWF provides 8 daily time horizon forecasts for each variable: 00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00, and 21:00 (8 is therefore the temporal resolution of the ensemble mean data). Therefore, there will be a maximum number of 8 observations per day in our dataset.

Table 1 Solar and wind energy meteorological input variables for quantile estimation

NWP variable	Usage
100 m u-component of wind	Solar & Wind
100 m v-component of wind	Solar & Wind
100 m wind norm	Wind
10 m u-component of wind	Wind
10 m v-component of wind	Wind
10 m wind norm	Wind
2 m temperature	Solar & Wind
Maximum 2 m temperature since previous postprocessing	Solar
Minimum 2 m temperature since previous postprocessing	Solar
Surface pressure	Solar & Wind
Mean surface downward longwave radiation flux	Solar
Mean surface downward shortwave radiation flux	Solar
Mean surface net longwave radiation flux	Solar
Mean surface net shortwave radiation flux	Solar
Mean top downward shortwave radiation flux	Solar
Mean top net longwave radiation flux	Solar
Mean top net shortwave radiation flux	Solar
Total cloud cover	Solar
Total precipitation	Solar

Usage column indicates whether the variable is used for solar energy, wind energy, or both

As previously explained, the independent variables for each observation (i.e., each row in the dataset) are obtained from ECMWF [47]. In addition, the dependent variable (electrical energy produced) is obtained from the ESIOS system [48] by matching the time horizons of each observation with the times from the ESIOS system (e.g., data from the 15:00 time horizon from ECMWF is matched with energy produced during the 15:00-16:00 time period from the ESIOS system). For wind energy, all forecast horizons are used. For solar energy, only those time horizons that correspond to year-round daylight hours (i.e., 09:00, 12:00, and 15:00) are used.

3 Methods

Given the independent variables $\mathbf{x} = (x_1, x_2, \ldots, x_p)$, the conditional distribution function (1) indicates the probability that the dependent variable Y is less than or equal to a given value. The α-quantile (2) is defined as the probability that Y is smaller than $Q_\alpha(\mathbf{x})$ is α.

$$F(y \mid \mathbf{X} = \mathbf{x}) = P(Y \leq y \mid \mathbf{X} = \mathbf{x}) \tag{1}$$

$$Q_\alpha(\mathbf{x}) = \inf\{y : F(y \mid \mathbf{X} = \mathbf{x}) \geq \alpha\} \tag{2}$$

In the following subsections, the machine learning methods used in this article to estimate the quantiles conditioned to the independent variables are described. In general, these quantile models will be represented by $\hat{Q}_\alpha(\mathbf{x})$. In these methods, a training set with N_{ins} instances $\mathcal{I}_{train} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{N_{ins}}, y_{N_{ins}})\}$ is used to fit $\hat{Q}_\alpha(\mathbf{x})$.

3.1 Linear quantile regression

The general framework of the linear quantile regression (LQR) model is derived from the linear regression model, which allows us to make predictions and inferences over the quantiles for some given dependent variables. Therefore, the α-quantile for a dependent variable is modeled as the linear combination of predictors:

$$\hat{Q}_\alpha(\mathbf{x}; \beta_0; \boldsymbol{\beta}) = \beta_0 + \boldsymbol{\beta}\mathbf{x} \tag{3}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ is the set of predictors, $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ is the set of coefficients and p is the number of predictors.

In contrast, classical linear regression models are built according to the minimization of the residuals from fitted values. Therefore, LQR has the quantile loss (or pinball loss) function (5) as the element to minimize, which is defined in terms of the residual (4). The LQR loss, which is asymmetrical, has a different penalty for residuals above ($u \geq 0$) or below ($u < 0$), and it can be shown that its minimization converges to the required α-quantile.

$$u = y - \hat{Q}_\alpha(\mathbf{x}) \tag{4}$$

$$L_\alpha(u) = \begin{cases} \alpha u, & u \geq 0 \\ (\alpha - 1)u, & u < 0 \end{cases} \tag{5}$$

Here, (5) is applicable for a single $(\mathbf{x}, y)$ pair, but generally, it is defined over a set of N_{ins} instances $T = \{(\mathbf{x}_i, y_i)_{i=1}^{N_{ins}}\}$, as shown in (6).

$$L_\alpha(T) = \frac{1}{N_{ins}} \sum_{i=1}^{N_{ins}} L_\alpha(y_i - \hat{Q}_\alpha(\mathbf{x}_i; \beta_0; \hat{\boldsymbol{\beta}})) \tag{6}$$

Thus, analogously to the linear model regression, the fitting process for LQR becomes a minimization process so that the parameters $\hat{\beta}$ can be obtained.

$$\underset{\beta_0, \hat{\boldsymbol{\beta}} \in R^{p+1}}{\text{minimize}} \sum_{i=1}^{N_{ins}} L_\alpha(y_i - \hat{Q}_\alpha(\mathbf{x}_i; \beta_0; \hat{\boldsymbol{\beta}})) \tag{7}$$

where N_{ins} is the size of the training data (i.e., the number of instances or observations). The LQR requires one model per quantile be trained: $\hat{Q}_{\alpha_1}, \hat{Q}_{\alpha_2}, \ldots, \hat{Q}_{\alpha_{N_{quan}}}$, where N_{quan} is the number of quantiles to be estimated.

During this study, the R package quantreg is implemented to fit the different LQR models and obtain an estimation of the conditional quantiles. Information about this package implementation can be found in [49].

3.2 Support vector quantile regression

Support vector quantile regression (SVQR) is a technique for estimating quantiles and is based on the idea of support vector regression (SVR) [50].

Standard SVR can be used for classification and regression. In the simplest approaches, linear models f are constructed, as shown in (8).

$$\hat{f}(\mathbf{x}) = \omega^T \mathbf{x} + b \tag{8}$$

where ω is a vector of weights, which are obtained by solving the minimization problem formulated in (9). This optimization process is utilized to find a balance between the simplicity of the model (the first term of (9)) and the loss of the model for each of the instances (the second term of (9)).

$$\underset{\omega, b}{\text{minimize}}\ \lambda ||\omega^T \omega + \frac{1}{N_{ins}} \sum_{i=1}^{N_{ins}} L(y_i - (\omega^T \mathbf{x}_i + b)) \tag{9}$$

where λ is a regularization hyperparameter that represents the tradeoff between these terms (sometimes C, or Cost, is used instead, where $C = \frac{1}{\lambda}$), and $L(u)$ is the loss function. For classification problems, the loss is usually the hinge loss, while for regression problems, the ϵ-insensitive L_1 loss is commonly used.

Nonlinear models $\hat{f}(\mathbf{x}) = \omega^T \chi(\mathbf{x}) + b$ can also be obtained by using nonlinear mappings χ. These nonlinear mappings are not explicitly applied. Instead, kernels and the kernel trick allow us to solve the optimization process required to train the SVR model without actually carrying out the mapping. The most widely used kernel is the radial basis function kernel (i.e., the Gaussian kernel), which is defined in (10). Nonlinear models can be defined in terms of the kernel in (11).

$$K_{RBF}(\mathbf{x_a}, \mathbf{x_b}) = \exp\left(-\frac{\|\mathbf{x_a} - \mathbf{x_b}\|^2}{2\gamma^2}\right) \tag{10}$$

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{i=N_{ins}} a_i K_{RBF}(\mathbf{x}, \mathbf{x}_i) \tag{11}$$

where a_i are coefficients obtained by the optimization process once the kernel has been included and γ is the kernel bandwidth parameter.

The concepts from SVR have been extended to quantile estimation [9] and used in recent work related to the energy field [13] by using the quantile loss L_α, which was defined in the previous section in (5), in the SVR optimization defined in (9). This extension allows us to use the SVR mechanism to extend quantile regression to nonlinear models. Given that the loss function L_α is different for different α values, a different model $\hat{Q}_\alpha$ has to be obtained for each α. The liquidSVM library is a recent and fast implementation of SVRs that provides methods for SVR-based quantile estimation [51] and is used for this study.

3.3 Gradient boosting quantile regression

Gradient boosting (GB) is an ensemble machine learning method. The GB models have the mathematical form shown in (12).

$$F_M(\mathbf{x}) = \sum_{j=1}^{j=M} \gamma_i h_i(\mathbf{x}) \tag{12}$$

where $h_i(\mathbf{x})$ are the members of the ensemble (called weak models) and $\gamma_i \geq 0$ are the weights of each model in the ensemble. M is the size of the ensemble (i.e., the total number of weak models).

The GB training method is sequential in the sense that a sequence of partial ensembles $F_1, F_2, \ldots, F_m, \ldots$ are constructed until the final ensemble F_M is obtained. This process is carried out by computing $F_{m+1}(\mathbf{x}) = F_m + \gamma_i h_m(\mathbf{x})$ so that F_{m+1} improves the previous ensemble F_m by adding a new ensemble member h_m. This process is repeated until the ensemble is complete.

Each new h_m model added to the ensemble is trained in a way that ensures that the transition from ensemble F_m to ensemble F_{m+1} follows a gradient descent procedure. This means that by adding h_m to ensemble F_m, the transition to F_{m+1} goes in the direction opposite that of the loss function gradient, i.e., in the direction in which the error decreases the most. This is achieved by training each h_m with a modified dataset, in which the inputs are the same as those in the original dataset, but the outputs are the negative gradients represented in (13).

$$r_i = -\frac{\partial(L(y_i, F(x_i))}{\partial F(x_i)} \,|_{F(x)=F_m(x)} \tag{13}$$

Thus, every h_m model added to the ensemble is trained with the $\{(\mathbf{x}_1, r_1), (\mathbf{x}_2, r_2), \ldots, (\mathbf{x}_{N_{ins}}, r_{N_{ins}})\}$ dataset. This general formulation of gradient boosting allows the method to optimize any loss function for which partial derivatives can be computed. Typically, loss functions such as the mean square error (MSE) or mean absolute error (MAE) are used, and this allows GB ensembles that optimize those loss functions to be obtained. However, this mechanism also allows us to obtain ensembles that optimize quantile loss, which is the function of standard gradient boosting software packages (for this article, LightGBM [52] is used). Given that different α values lead to different quantile loss functions, a different ensemble has to be trained for every α-quantile.

In this section, the main ideas of GB as applied to quantile regression have been illustrated. Other technical details have not been discussed, but a complete overview of GB can be found in [11]. Additionally, although in principle the ensemble member h_i can be any kind of model, most implementations have used regression trees as base models, which have been shown to be very powerful and efficient approaches. Finally, in this study, we have used the LightGBM implementation, which has its advantages and technical issues. While slightly different to the foundational ideas discussed in this section, LightGBM can be examined in [52].

The main hyperparameters of GB are the number of ensemble members M, the maximum depth of the trees in the ensemble, and the shrinkage (or learning rate) ν. If a learning rate different than 1.0 is used, then the GB ensemble becomes (14). All these hyperparameters allow us to regularize the ensemble and control overfitting. Large M values, large maximum depth, or large learning rates usually lead to overfitting, and their values must be carefully adjusted so that models with good generalization are obtained.

$$F_M(x) = \sum_{j=1}^{j=M} \nu\gamma_i h_i(x) \tag{14}$$

Similarly to LQR, GBQR requires that one model per quantile be trained: $\hat{Q}_{\alpha_1} = F_{\alpha_1,M}$, $\hat{Q}_{\alpha_2} = F_{\alpha_2,M}, \ldots$

3.4 Natural gradient boosting

Natural gradient boosting (NGBoost) is a recent method that uses boosting models for computing probabilistic predictions in regression problems [12, 16, 53]. The first difference between NGBoost and standard boosting is that the ensemble model is used in NGBoost to estimate the parameters of the conditional probability distribution (e.g., the mean μ and standard deviation $log(\sigma)$ of the normal distribution $f_{(\mu,\sigma)}(y|X=x)$) rather than the dependent variable Y. In other words, the output(s) of the boosting ensemble described in 12 are the parameters of the probability distribution for the dependent variable and not the dependent variable itself. For instance, if the parameters are μ and $\log(\sigma)$, a model with two ensembles, one ensemble per parameter, are obtained (see 15). Quantiles can be then obtained from these probability distributions (namely, $N(F_M^{(\mu)}(\mathbf{x}), \exp(F_M^{(\log(\sigma))}(\mathbf{x}))$, where N is the normal distribution).

$$\begin{aligned} \hat{\mu} &= F_M^{(\mu)}(\mathbf{x}) = \sum_{j=1}^{j=M} \gamma_i h_i^{(\mu)}(\mathbf{x}) \\ \log\hat{(}\sigma) &= F_M^{(\log(\sigma))}(\mathbf{x}) = \sum_{j=1}^{j=M} \gamma_i h_i^{(log(\sigma))}(\mathbf{x}) \end{aligned} \tag{15}$$

The second difference between NGBoost and standard boosting is that rather than using the standard gradient, as shown in 13, NGBoost uses the natural gradient. The reason is that to obtain gradients for this formulation, distances between different probability distributions must be computed. However, the distances between the parameters that represent distributions (e.g. (μ, $\log(\sigma)$)) do not represent the differences between their associated probability distributions well. Thus, natural gradients, which use divergences such as the Kullback·Leibler divergence or the L^2 divergence are defined as the proper way to consider the differences between the actual probability distributions. Natural gradients are used instead of standard gradients for the GB algorithm.

3.5 Quantile regression forests

Random forests (RFs) are another ensemble machine learning method. Unlike gradient boosting, the ensemble in RFs is not based on the improvement of a weak learner; instead, it is based on fitting a large number of learners and bagging to make a joint prediction.

One of the particularities of this method is that it relies on randomization to prevent overfitting. From training data

$\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{N_{ins}}, y_{N_{ins}})\}$ of size N_{ins}, each one of the M base learners $\{h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_M(\mathbf{x})\}$ (regression trees for this project) takes a bootstrapped sample with replacement. Furthermore, only a random subset of m features from the p available features are employed to grow the trees. Trees are grown until the minimum sample size required for splitting a node is reached [54]. The number of trees M, the maximum number of selected features m, and the minimum number of observations required to split a node of the tree are important hyperparameters of this method.

Following [10], predictions are made using standard random forests by averaging the individual predictions of each of the trees in the ensemble ($\{h_1, h_2, \ldots, h_M\}$). Each tree h_i makes a prediction by sending a new instance $\mathbf{x}$ down the tree until it reaches a leaf. The leaf contains all the observations $\{(\mathbf{x}_i, y_i)\}$ that reached it during the training process. The prediction of the forest is simply the average of the dependent variable of those instances ($\hat{y}(\mathbf{x}) = \frac{1}{M}\sum_{j=1}^{j=M} h_j(\mathbf{x})$).

This process can be used for point prediction, and random forests can easily be used for estimating quantiles [10]. Given that the leaf reached by a new instance $\mathbf{x}$ contains a set of observations, $\{(\mathbf{x}_i, y_i)\}$, $\{y_i\}$ can be used for constructing an empirical distribution. These empirical distributions can be averaged across all trees in the ensemble. From this average distribution, quantiles can be computed.

More formally, let:

- $l(\mathbf{x}, h_j)$ be the leaf of ensemble tree h_j, which is reached by new instance $\mathbf{x}$.
- $T(\mathbf{x}, h_j)$ be the set of training instances $\{(\mathbf{x}_i, y_i)\}$ that reach leaf $l(\mathbf{x}, h_j)$ during the training process.
- $w(\mathbf{x}, h_j, y) = \frac{|\{(\mathbf{x}_i, y_i)\} \in T(\mathbf{x}, h_j)_{|y_i=y}|}{|T(\mathbf{x}, h_j)|}$ be the proportion of instances in $T(\mathbf{x}, h_j)$ for which $y_i = y$. If no instance in $T(\mathbf{x}, h_j)$ has the value y for the dependent variable, then $w(\mathbf{x}, h_j, y) = 0$
- $w(\mathbf{x}, y) = \frac{1}{M}\sum_{j=1}^{M} w(\mathbf{x}, h_j, y)$ be the average of w across all M trees in the random forest ensemble.

The final conditional distribution function can be estimated by the empirical distribution of the unique values $y_i \in L(\mathbf{x}, h_j)$, assuming that each value has probability $w(\mathbf{x}, y_i)$, as can be seen in (16).

$$\hat{F}(y \mid X = \mathbf{x}) = \sum_{i=1}^{uv} w(\mathbf{x}, y_i) 1_{y_i \leq y} \tag{16}$$

where uv is the number of unique values of the dependent variable present in leaves $l(\mathbf{x}, h_j)$ and $\{y_1, y_2, \ldots, y_{uv}\}$ are the unique values. Unlike LQR and GBQR, QRF allows the extraction of all desired quantiles $(\alpha_1, \alpha_2, \ldots, \alpha_{N_{quan}})$ from a single model.

During the development of this article, the scikit-garden in Python was implemented to fit the different QRF models [55].

3.6 Quantile regression deep neural networks

Neural networks have been proven to be powerful methods for both classification and regression. In this work, DNNs are used to estimate a set of quantiles, and the model named QRDNN (see Fig. 2) has been introduced. Like most fully-connected DNNs, QRDNN can be visualized with an input layer, which contains predictors or inputs $\mathbf{x}$, several hidden layers, where each layer has a defined number of neurons, and an output layer, which, in this work, are the estimated quantiles.

The operation of the hidden layers can be understood as matrix multiplication followed by a nonlinear activation function g (e.g., ELU, ReLU or sigmoid). If $\mathbf{x}$ is the vector of inputs, $\mathbf{L_1}$ is the weight matrix from the input layer to the first layer, and $\mathbf{b_1}$ is the vector of biases from the first hidden layer. Then, the output of the first layer is given by (17).

$$\mathbf{a_1} = g(\mathbf{L_1 x} + \mathbf{b_1}) \tag{17}$$

With the exception that the activation of the previous layer is utilized, the same structure is followed for the remaining hidden layers until the output layer is reached. Thus, the output of the i-th layer (i=2,3,...) is given by (18).

$$\mathbf{a_i} = g(\mathbf{L_i a_{i-1}} + \mathbf{b_i}) \tag{18}$$

where $\mathbf{L_i}$ is the weight matrix from the layer $i - 1$ to layer i and $\mathbf{b_i}$ is the bias vector of layer i.

The outputs of the neural network (activation of the output layer) are the estimated quantiles and the network will have one neuron output for each α-quantile to be estimated, as can be seen in Fig. 2.

Training large neural networks that contain several hidden layers with many neurons in each layer may lead to overfitting. A common approach to prevent overfitting is to use dropout layers. These additional layers randomly hide or ignore some outputs from a hidden layer with a probability p. Thus, the DNN will not employ all weights. Therefore, it is more difficult to overfit the training data, which results in a network with better generalization.

Loss functions usually used for training neural networks are the mean square error (for regression) and cross-entropy (for classification). However, when the neural network is used to estimate quantiles, these functions are not useful. Given that quantile estimation can be formulated as the minimization of quantile loss ((5) and (6)), the approach followed in this work is based on the optimization of these functions.

However, instead of using (5) and (6) in a straightforward way, (19), an equivalent formulation, is used. The reason

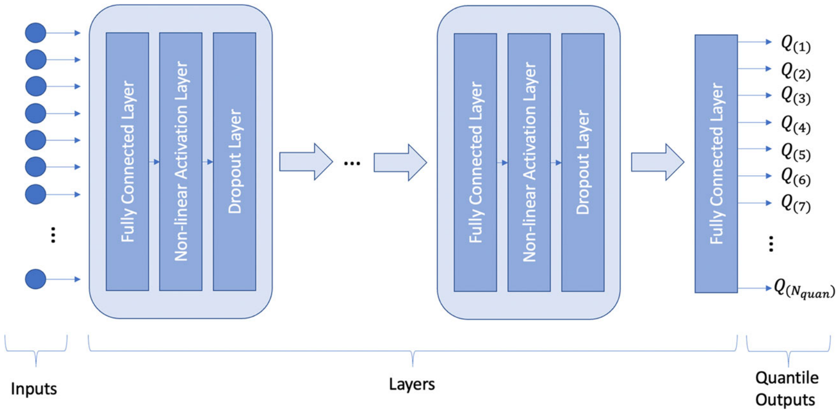


Fig. 2 QRDNN architecture to estimate N_{quant} quantiles

is that a straightforward implementation of (5) and (6) would require a loop over the instances, where for every instance, a check on whether the residual (4) is positive or negative must be completed, and then αu and $(\alpha - 1)u$ can be computed. In (19), the explicit loop is removed, which allows for a more efficient execution when using PyTorch [56] and graphical processing units (GPUs).

$$L_\alpha(T) = \frac{1}{N_{ins}} \sum \max(\alpha U_\alpha, (\alpha - 1)U_\alpha) \tag{19}$$

where $U_\alpha = (u_{\alpha,1}, u_{\alpha,2}, \ldots, u_{\alpha,N_{ins}})^T$ is a column vector containing the residuals $u_{\alpha,i} = y_i - \hat{Q}_\alpha(\mathbf{x}_i)$ for all instances in the training set. max returns a column vector $(\max(\alpha u_{\alpha,1}, (\alpha - 1)u_{\alpha,1}), \max(\alpha u_{\alpha,2}, (\alpha - 1)u_{\alpha,2}), \ldots)^T$. Given that $0 < \alpha < 1$ and $(\alpha - 1)$ is always negative, max will return $\alpha u_{\alpha,i}$ if the residual u_i is positive, and $(\alpha - 1)u_{\alpha,i}$ otherwise. Hence, it is equivalent to (5). $\sum$ represents the addition of all elements in the vector.

Deep networks have several hyperparameters that are important to tune. In this work, these are:

- The number of layers, and the number of neurons per layer. If the model is too complex, there is a risk of overfitting in the network, but if the model is too simple, underfitting might occur.
- The learning rate. The learning rate controls the size of the learning step. If it is too large, the optimum can be missed.
- The batch size. Generally, the loss and parameter updates are completed in packets called minibatches, which are smaller than the complete dataset. Finding the right minibatch size can be important.
- Activation layer. Different nonlinear layers may work better for particular problems; hence, it is important to find the right one. In this article, sigmoid, tanh, ELU and ReLU are tested. The ELU, which has a parameter α that controls its shape, is a (soft) alternative to ReLU.
- Optimizer. Whereas stochastic gradient descent (SGD) is the most widely used optimizer, for some problems, better results may be obtained using advanced optimizers. In this article, we also test Adam, an optimizer that is well-known for its excellent results [57].

To program the neural networks for this work, the PyTorch framework is used [56].

4 Methodology

4.1 Models: conditional quantiles and prediction intervals

In this article, the model $\hat{Q}_\alpha(\mathbf{x})$ takes inputs $\mathbf{x}$ (i.e., the independent variables) and returns the conditional α-quantile for the inputs. Some methods (e.g., QRF and QRDNN) can return multiple quantiles $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_{N_{quan}}\}$ from a single model $\hat{Q}_{\boldsymbol{\alpha}}$.

For QRDNN, better results may be achieved in terms of quantile loss when training only one conditional quantile rather than training a set of quantiles. However, this requires training one deep neural network for every conditional quantile. As the goal of this article is to propose an efficient method where several PIs can be built from the multiple-quantile output, training QRDNN with a set of $\boldsymbol{\alpha}$ quantiles is preferred.

The inputs $\mathbf{x}$ of the model are the selected meteorological variables on the grid that cover the regions of interest. For instance, for the Lugo region, a grid of size 5 × 4 is defined (see Fig. 1(a)). Given that Lugo is a wind region, and 8 meteorological variables have been selected for wind, the model will have $5 \times 4 \times 8 = 160$ variables. For Córdoba, which is a solar region, $\mathbf{x}$ will contain $5 \times 4 \times 15 = 300$ meteorological variables.

In this article, conditional PIs are also constructed from the (conditional) quantiles. A conditional PI for inputs $\mathbf{x}$ is a pair of lower and upper bounds that contain the dependent variable with a probability called the prediction interval nominal probability (PINP). Alternatively, the probability of not covering the dependent variable can also be used. Note that in other works, α is used to represent this probability. However, in this work, α represents the α-quantiles. Therefore, in this study, this probability will be referred to as $\varepsilon = 1 - PINP$. PIs can be computed by using quantiles $\frac{\varepsilon}{2}$ and $1 - \frac{\varepsilon}{2}$ as lower and upper bounds, respectively. Using these quantiles, a probability of $\frac{\varepsilon}{2}$ remains uncovered to the left of the lower bound and $\frac{\varepsilon}{2}$ remains uncovered to the right of the upper bound. This type of interval covers exactly $1 - (\frac{\varepsilon}{2} + \frac{\varepsilon}{2}) = 1 - \varepsilon = PINP$.

$$\begin{aligned}\hat{PI}_{1-\varepsilon}(\mathbf{x}) &= \left[\hat{Low}_{\varepsilon}(\mathbf{x}), \hat{Upp}_{\varepsilon}(\mathbf{x})\right] \\ &= \left[\hat{Q}_{\frac{\varepsilon}{2}}(\mathbf{x}), \hat{Q}_{1-\frac{\varepsilon}{2}}(\mathbf{x})\right] \qquad (20)\end{aligned}$$

For instance, a 99% prediction interval can be built as shown in (21).

$$\hat{PI}_{0.99}(\mathbf{x}) = \left[\hat{Q}_{.005}(\mathbf{x}), \hat{Q}_{.995}(\mathbf{x})\right] \qquad (21)$$

4.2 Evaluation procedure

To train and evaluate the models, three datasets are constructed: the training, validation, and test sets. Two full years of data are used for the training set, one different year is used for the validation set and hyperparameter tuning, and another year is used for the test set. A 4-year period is selected so that the maximum generation remains approximately constant for the whole period. As a result, models that are trained using some years can be tested without having to adapt the remaining years.

The datasets created for each of the four Spanish regions considered in this work are described below:

- Lugo (wind energy). Training set: years 2015 and 2016. Validation set: year 2017. Test set: year 2018. A total of 160 inputs (20 grid points times 8 NWP variables).
- Granada (wind energy). Training set: years 2015 and 2016. Validation set: year 2017. Test set: year 2018. A total of 192 inputs (24 grid points times 8 NWP variables).
- Córdoba (solar energy). Training set: years 2016 and 2017. Validation set: year 2018. Test set: year 2019. A total of 300 inputs (20 grid points times 15 NWP variables).
- Ciudad Real (solar energy). Training set: years 2015 and 2016. Validation set: year 2017. Test set: year 2018. A total of 270 inputs (18 grid points times 15 NWP variables).

All independent variables in the three sets were standardized by computing the required standard deviation and mean of the training and validation sets for each region and using them on the training, validation, and test partitions. Concerning the dependent variable, some transformations were applied to address normality issues. A decimal logarithm transformation was applied and was followed by a standardization using the same procedure as that used for the independent variables. In Fig. 3, the transformation process of the dependent variable can be seen. In Fig. 3 (a) and (c), the histograms of the dependent test variable are shown for Granada (wind energy) and Ciudad Real (solar energy), respectively. The Ciudad Real dependent variable histogram, which has an almost bimodal distribution (i.e., small and large amounts of energy generated), suffers from a larger shape change. In Fig. 3, (3(b) and (d)), the histograms of the transformed dependent variable are presented.

This transformation allows us to reduce the skewness of the distribution.

Standardizing the complete dataset may potentially improve the training process as both dependent and independent variables have the same range of values and similar shapes. In addition, some model weights will no longer dominate others.

In the training process, 10 quantiles are modeled by each method for every region. These quantiles are given as follows: $Q_{.005}$, $Q_{.025}$, $Q_{.05}$, $Q_{.075}$, $Q_{.1}$, $Q_{.9}$, $Q_{.925}$, $Q_{.95}$, $Q_{.975}$ and $Q_{.995}$. This enables the possibility of building 5 PIs that have different coverage: $\text{PI}_{80\%}$, $\text{PI}_{85\%}$, $\text{PI}_{90\%}$, $\text{PI}_{95\%}$ and $\text{PI}_{99\%}$.

To select the best possible combination of hyperparameters, an exhaustive grid search is completed. We explore all possible combinations of hyperparameter values within a predefined space. The sets of values are presented in Table 2.

It is important to note the differences between the methods used in the fitting process. While LQR and GBQR can obtain one conditional quantile per model, QRF, NGB, and SVQR can fit the complete conditional distribution function and QRDNN can obtain all ten quantiles at once by means of the set structure.

The evaluation procedure has been developed by choosing the hyperparameter set with the smallest average quantile loss across the ten selected quantiles (24). Thus, we extract the 10 conditional quantiles from the methods and calculate the mean quantile loss across them for all the hyperparameter values. Therefore, this means that GBQR is modeled with the same hyperparameter configuration for all the quantiles so that a homogeneous selection can be obtained.

Thus, the best hyperparameter values for each method, region and type of energy (wind/solar) are given in Table 3.

In some methods, such as LQR, GBQR and QRDNN, predicting close multiple conditional quantiles may introduce the problem of quantile crossing. This may specifically occur when quantiles are very close (e.g. $Q_{0.975}$ and $Q_{0.995}$). To solve this problem and when evaluating the models on

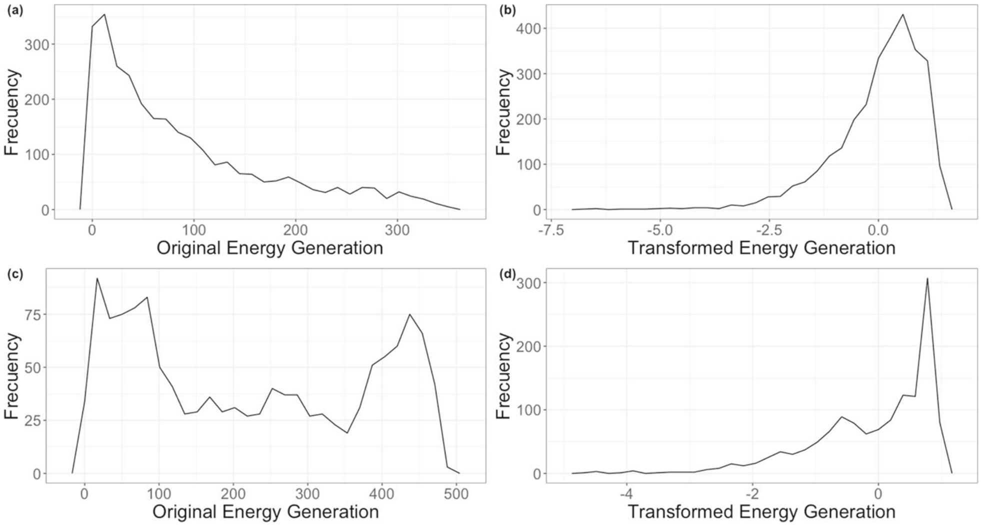


Fig. 3 (a) Histogram of the generated wind energy in Granada during 2018, and (b) histogram after transformation (of the dependent variable). (c) Histogram of the generated solar energy generated in Ciudad Real during 2018, and (d) histogram after transformation (of the dependent variable). After a logarithmic transformation is applied, data are standardized by subtracting the mean and scaling by the standard deviation

the test sets, model predictions (i.e., the list of quantiles) are sorted in ascending order.

4.3 Metrics

During the development of this work, several metrics were employed to evaluate the different models. First, quantile loss was used to evaluate models on the test set and to select the best performing model during the hyperparameter tuning on the validation set. This metric was already defined in (4), (5), and (6), but it is reproduced in (22) and (23) below for convenience.

Table 2 Hyperparameter values explored during the grid search for each method

Method	Hyperparameter space
LQR	-
GBQR & NGB	Learning rate: $j \times 10^{-i}$, where $j \in \{1, 5\}$ and $i \in \{1, 2, 3, 4\}$ Number of trees: 500 to 5000 in steps of 500 Max depth: 2 to 14 in steps of 2
SVQR	Cost (C): 0.1, 0.5, 1, 10, & 50 to 500 in steps of 50 Gamma (γ): 0.1, 0.5, 1, 10, & 50 to 500 in steps of 50
QRF	Min samples for splitting: 5, 10, 20, 30, 40, 50, & 100 Number of trees: 10, 50, 75, & 100 to 700 in steps of 100 Max features: 10%, 20%, ..., 100% of possible attributes
QRDNN	Hidden layers: 1,2,3,...,10 Neurons per layer: 50, 100, 150, 200, &250 Learning rate: $j \times 10^{-i}$, where $j \in \{1, 5\}$ and $i \in \{1, 2, 3, 4, 5\}$ Batch size: 2^i, where $i \in \{4, 5, 6, ..., 10\}$ Optimizers: SGD & Adam Activation layers: sigmoid, tanh, ELU & ReLU

$$L_\alpha(u) = \begin{cases} \alpha u, & u \geq 0 \\ (\alpha - 1)u, & u < 0 \end{cases} \tag{22}$$

$$L_\alpha(T) = \frac{1}{N_{ins}} \sum_{i=1}^{N_{ins}} L_\alpha(y_i - \hat{Q}_\alpha(\mathbf{x}_i)) \tag{23}$$

where $T = \{(\mathbf{x}_1, y_1), \ldots\}$ is a test (or validation) set with N_{ins} instances.

In general, we are interested in obtaining models not just for a specific quantile α but for a set of quantiles $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_q, \ldots, \alpha_{N_{quan}}\}$. In this case, the average quantile loss across all different quantiles can be used.

$$L_{\boldsymbol{\alpha}}(T) = \frac{1}{N_{quan}} \sum_{q=1}^{N_{quan}} L_{\alpha_q}(T) \tag{24}$$

The continuous ranked probability score (CRPS) is a metric that measures the quality of a probability distribution [58]. When the distribution is represented by multiple quantiles, as it is in our case, CRPS is defined by (25).

$$\begin{aligned} & CRPS(T) \\ & = \frac{1}{N_{ins}} \sum_{i=1}^{N_{ins}} \left(\frac{1}{N_{quan}} \sum_{k=1}^{N_{quan}} \mid \hat{Q}_{\alpha_k}(\mathbf{x}_i) - y_i \mid \right. \\ & \left. - \frac{1}{2N_{quan}^2} \sum_{k=1}^{N_{quan}} \sum_{l=1}^{N_{quan}} \mid \hat{Q}_{\alpha_k}(\mathbf{x}_i) - \hat{Q}_{\alpha_l}(\mathbf{x}_i) \mid \right) \end{aligned} \tag{25}$$

It can be seen that CRPS is the addition of two components. The first component measures the distance between each of the quantiles and the actual value of the dependent variable. The value of this component will be minimized when the quantiles accurately reflect the data distribution.

The second component, which is independent of the data, measures the distance between the quantiles. The minimization of this component leads to sharper distributions (i.e. quantiles are closer to each other). The lower the CRPS is, the better. In fact, when quantile predictions degenerate to point predictions (i.e. all quantiles become the same value, and a single prediction is produced), CRPS becomes the mean absolute error (MAE).

Other metrics have been used in this work to evaluate PIs. The prediction interval coverage probability (PICP) [59] measures the proportion of instances covered by the interval, and it is given by (26).

$$PICP = \frac{1}{N_{ins}} \sum_{i=1}^{N_{ins}} \mathbf{1}_{y_i \in \hat{PI}(\mathbf{x}_i)} \tag{26}$$

where $\mathbf{1}_{y_i \in \hat{PI}(\mathbf{x}_i)}$ is an indicator function whose value is 1 when $y_i \in \hat{PI}(\mathbf{x}_i)$ for a given $\mathbf{x}_i$ and 0 otherwise. $\hat{PI}(\mathbf{x}_i)$ is the prediction interval associated with instance $\mathbf{x}_i$. The PICP is expected to be larger than the actual probability, which is known as the prediction interval nominal probability (PINP), but should be as close to it as possible.

Another important metric is the width of the generated intervals. The average interval width (AIW) [59] is shown in (27) and it is normalized for the maximum possible width of every set.

$$AIW = \frac{1}{N_{ins}(y_{\max} - y_{\min})} \sum_{i=1}^{N_{ins}} \hat{Upp}(\mathbf{x}_i) - \hat{Low}(\mathbf{x}_i) \tag{27}$$

where $\hat{Upp}(\mathbf{x}_i)$ and $\hat{Low}(\mathbf{x}_i)$ are the upper and lower bounds of the prediction interval for $\mathbf{x}_i$, respectively.

Given that it is trivial to attain high coverage by increasing the interval width, a simple but effective metric is the ratio between the coverage and width [15], as shown in (28). When there are similar PICPs among different models, a larger ratio provides a better understanding of model performance.

$$R_{c-w} = \frac{PICP}{AIW} \tag{28}$$

The Winkler score (WS) (see (29)) is a widely used metric to evaluate PIs. It is basically the width of the PI with an added penalty for those observations outside the interval bounds [60]. Therefore, the smaller the WS is, the better.

$$W_{i,\varepsilon} = \begin{cases} (\hat{Upp}_\varepsilon(\mathbf{x}_i) - \hat{Low}_\varepsilon(\mathbf{x}_i)) + \frac{2}{\varepsilon}(\hat{Low}_\varepsilon(\mathbf{x}_i) - y_i), & \text{if } y_i < \hat{Low}_\varepsilon(\mathbf{x}_i) \\ (\hat{Upp}_\varepsilon(\mathbf{x}_i) - \hat{Low}_\varepsilon(\mathbf{x}_i)), & \text{if } \hat{Low}_\varepsilon(\mathbf{x}_i) \le y_i \le \hat{Upp}_\varepsilon(\mathbf{x}_i) \\ (\hat{Upp}_\varepsilon(\mathbf{x}_i) - \hat{Low}_\varepsilon(\mathbf{x}_i)) + \frac{2}{\varepsilon}(y_i - \hat{Upp}_\varepsilon(\mathbf{x}_i)), & \text{if } y_i > \hat{Upp}_\varepsilon(\mathbf{x}_i) \end{cases} \tag{29}$$

Table 3 Best hyperparameter values selected by grid search for each method and region

Method	Hyperparameter	Lugo (wind)	Granada (wind)	Córdoba (solar)	C. Real (solar)
GBQR	Learning rate	1×10^{-3}	1×10^{-2}	1×10^{-3}	5×10^{-4}
	Number of trees	5000	1000	5000	4500
	Max depth	2	2	2	4
NGB	Learning rate	1×10^{-4}	5×10^{-4}	1×10^{-4}	1×10^{-4}
	Number of trees	3000	1500	4500	3500
	Max depth	6	4	8	6
SVQR	Cost	50	50	100	100
	Gamma	200	150	300	500
QRF	Min obs. for splitting	10	5	5	10
	Number of trees	600	300	600	100
	Max features	10%	50%	10%	10%
QRDNN	Hidden layers	7	5	4	5
	Neurons per layer	50	200	250	150
	Learning rate	1×10^{-6}	1×10^{-6}	1×10^{-6}	1×10^{-6}
	Batch size	2^9	2^9	2^8	2^8
	Optimizer	Adam	Adam	Adam	Adam
	Activation layers	ELU($\alpha = 1$)	ELU($\alpha = 1.5$)	ELU($\alpha = 1$)	ELU($\alpha = 1$)

where $\hat{\text{Upp}}_{\varepsilon}(\mathbf{x}_i)$ and $\hat{\text{Low}}_{\varepsilon}(\mathbf{x}_i)$ represent the upper and lower bounds of the interval for $\mathbf{x}_i$, and ε is defined for the PIs by $(1-\varepsilon) = PINP$ the desired coverage. W_{ε} is obtained as the average of the $W_{i,\varepsilon}$ over all the instances in a test set.

4.4 Conformalized quantile regression for prediction interval estimation

As seen in Section 4.1, the properties of the associated prediction interval, such as the coverage or width, are not directly take into account when constructing PIs from estimated conditional quantiles. In other words, we rely on a good estimation of the quantiles, but the PI itself is not directly optimized.

To consider these properties in our estimated PIs, we apply conformalized quantile regression (CQR) [41] in our methodology. The CQR framework is based on the posterior adjustment of conditional quantiles by means of a validation set. This has been recently applied to wind power estimation in a time series context with good results [42].

Let $\hat{Q}_{\alpha}(\mathbf{x})$ be a model obtained from training set $\mathcal{I}_{train}$ for estimating two quantiles to construct a PI with a target coverage of $1-\varepsilon = PINP$, as depicted in (30).

$$\left[\hat{Q}_{\frac{\varepsilon}{2}}, \hat{Q}_{1-\frac{\varepsilon}{2}}\right] \leftarrow \hat{Q}\left(\{\mathbf{x}_i, y_i\} : i \in \mathcal{I}_{train}\right) \tag{30}$$

Then, conformity scores are computed by evaluating PIs on the validation set $\mathcal{I}_{val}$:

$$E_i := \max\left\{\hat{Q}_{\frac{\varepsilon}{2}}(\mathbf{x}_i) - y_i,\ y_i - \hat{Q}_{1-\frac{\varepsilon}{2}}(\mathbf{x}_i)\right\}, \quad i \in \mathcal{I}_{val} \tag{31}$$

This score represents the distance from the value y_i to the PI when the target value is not covered by the PI and the maximum distance to one of the PI bounds when the PI includes the target variable. Therefore, this score considers both undercoverage and overcoverage.

Finally, we can build a PI with calibrated quantiles for y_{i+1} from data $\mathbf{x}_{i+1}$ as

$$\left[\hat{Q}_{\frac{\varepsilon}{2}} - q_{1-\varepsilon}(E, \mathcal{I}_{val}),\ \hat{Q}_{1-\frac{\varepsilon}{2}} + q_{1-\varepsilon}(E, \mathcal{I}_{val})\right] \tag{32}$$

where $q_{1-\varepsilon}(E, \mathcal{I}_{val})$ represents the $(1-\varepsilon)$-th empirical quantile of $\{E_i : i \in \mathcal{I}_{val}\}$. The PIs constructed from calibrated quantiles are supposed to better approach the PINP, reducing their width in case of overcoverage and increasing it in case of undercoverage.

We note that this is a general approach. In our problem, we estimate PIs for different PINPs and time horizons. Therefore, we propose adapting the CQR methodology by computing different conformity scores for each PINP and time horizon considered. Therefore, the resulting calibrated PI for a specific PINP and time t is shown in (33).

$$\left[\hat{Q}_{\frac{\varepsilon}{2}} - q_{1-\varepsilon}(E_{1-\varepsilon,t}, \mathcal{I}_{val,t}),\ \hat{Q}_{1-\frac{\varepsilon}{2}} + q_{1-\varepsilon}(E_{1-\varepsilon,t}, \mathcal{I}_{val,t})\right] \tag{33}$$

Note that $\mathcal{I}_{val,t}$ is the validation set, but only for observations at time t. Overall, this approach is useful for solar energy regions, where PIs present more differences depending on the time horizon.

5 Results

In this section, we discuss the results obtained on the test sets. First, model performance is evaluated in terms of the accuracy of the quantiles. Next, PIs are built from the quantiles and tested according to their coverage, width, and WS. Then, PIs are estimated from the calibrated quantiles as explained in Section 4.4 to show their improvement. Finally, an analysis by season is presented for the PIs generated by the QRDNN.

5.1 Quantile estimation

We present the average quantile loss (Fig. 4) obtained by the 10 quantiles and report the results by the time horizon (hours), method, and region. Results have also been averaged across all time horizons, as displayed in the rightmost columns of Fig. 4.

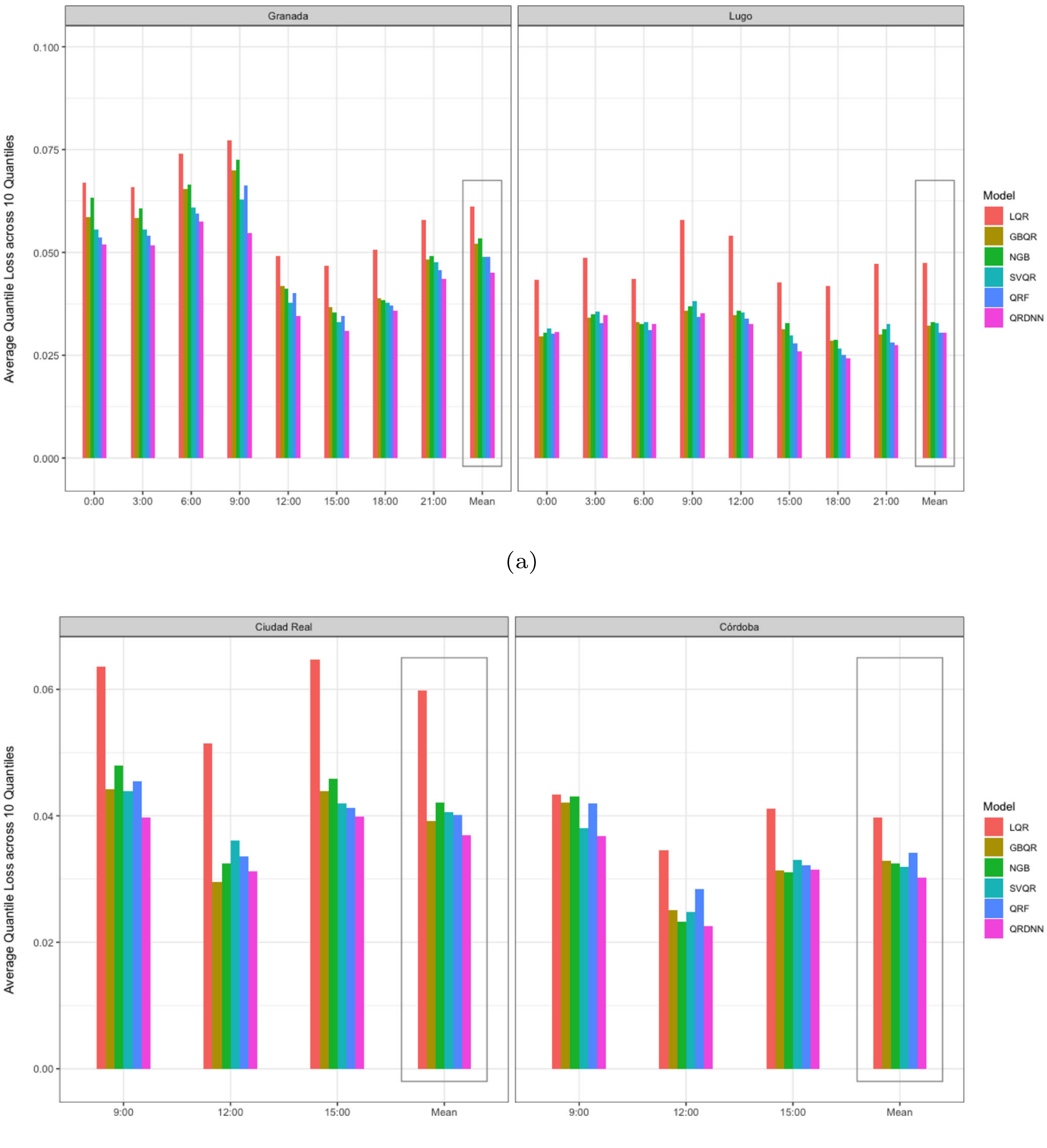
Regarding wind energy forecasting (Fig. 4(a)), the largest loss in all cases is observed when LQR is used. The performance of NGB is usually the second worst, followed by GBQR. Regarding the best performing methods, the best results on the Lugo data are achieved using QRDNN and QRF, whereas on the Granada data, the best results are achieved using QRDNN and SVQR; slightly less loss is observed for the majority of the time horizons and also on average for QRDNN. Furthermore, we note that the four methods perform better on the Lugo data than on the Granada data.

With respect to solar energy forecasting, as shown in Fig. 4(b), the largest loss for every time horizon is always observed when using LQR. On average, NGB and QRF are the second worst performing methods on the Ciudad Real and Córdoba data, respectively. The most accurate method is again QRDNN, where slightly less loss is observed except for on the Ciudad Real data at 12:00.

Another metric that gives a general understanding of the method performance regarding the accuracy of quantiles is CRPS (Fig. 5). It can be seen that the results are similar to those of quantile loss: the lowest loss is mainly obtained using QRDNN, both for solar and wind energy predictions.

However, there are some changes in the rest of the methods. For example, a low CRPS is obtained using LQR for solar energy prediction (Fig. 5(b)). However, we will later see that this result comes at the cost of low coverage.

Favorable performance is not obtained for this metric when QRF is used because in solar energy prediction,

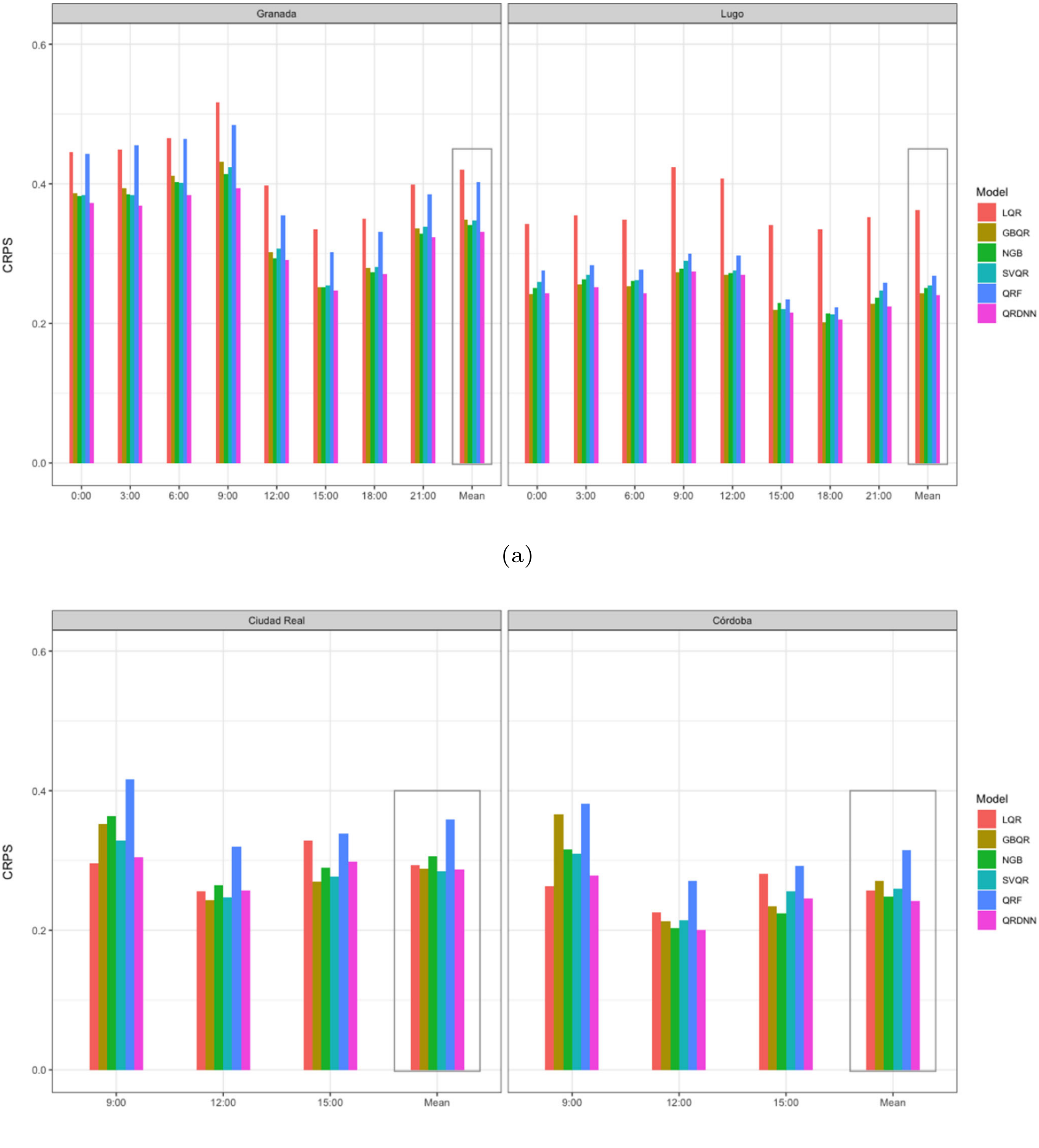


(a)

(b)

Fig. 4 Average quantile loss for the different methods based on the time horizon. (a) Wind energy regions and (b) solar energy regions. The rightmost column shows the average across all time horizons. On average, the best results are obtained using QRDNN, except on the Lugo data, where similar results for QRDNN and QFR are achieved

the worst CRPS values are obtained. For wind energy prediction, QRF has worse results than those obtained using GBQR. Similar CRPS values are achieved using NGB and SVQR in the wind energy regions. In solar regions, SVQR is slightly better than NGB on the Ciudad Real data, and the opposite behavior is observed on the Córdoba data.



(a)

(b)

Fig. 5 CRPS values obtained by the different methods based on the time horizon. (a) Wind energy regions and (b) solar energy regions. The rightmost column is the average across all time horizons. On average, the best results are achieved using QRDNN, except on the Ciudad Real data, where similar performances are obtained for QRDNN, GBQR and SVQR

5.2 Prediction interval estimation

Methods studied in this work are used to build PIs from their estimated quantiles, as described in Section 4. Therefore, the PI metrics PICP and AIW are presented for the regions of Granada, Lugo, Ciudad Real and Córdoba in Tables 4, 5, 6 and 7, respectively.

Each of these tables contains 5 subtables, one per PINP target value. There is one row per method and one column per time horizon. The table cells show both the PICP value and AIW value (the latter is shown in parentheses). The rightmost column is the average of the PICP and AIW values (the latter is shown in parentheses) across all time horizons. Note that in all the resulting tables, when a PICP equal to or higher than the target PINP is achieved for a given method, the corresponding value is shown in bold.

First, the PICP and AIW results from Granada (wind energy) are presented in Table 4. Generally, the desired coverage is not able to be achieved using LQR, GBQR, and NGB, whereas it can be achieved using SVQR on a few occasions. In contrast, reasonable coverage is obtained using QRF and QRDNN at most of the times and on average (rightmost column). Coverage is achieved for all PIs at all hours using QRF, except at 09:00. However, the desired coverage is not obtained in some cases using QRDNN: this mostly occurs in the first half of the day (00:00, 03:00, 06:00, 09:00) and also for high PINP values. However, it is important to note that the difference between PICP and PINP is quite small in these cases. In contrast, the AIW is the smallest among the rest of the methods for every target PINP and at hour analyzed. Additionally, in terms of the mean (rightmost column of Table 4), only when using QRDNN and QRF can the target PINP (or close to it) be obtained, but the narrowest intervals (smallest AIW) are obtained using QRDNN.

In Table 5, the results on the Lugo data (wind energy) are shown. A similar behavior, one in which the desired coverage is not reached, is observed for LQR and GBQR. Except for the PINP at 99%, coverage is also not obtained using SVQR. In this region, the performance of NGB is improved and the method is able to be used to achieve the desired coverage (on average) for PINP at 80%, 85%, and 90% with a relatively low interval width, whereas it is achieved using QRF for all PINP values and times. The PINP in all cases for the 80% and 85% PIs, and in most cases for the 90% and 95% PIs are met using QRDN. Nevertheless, the only PINP where coverage is not reached on average (rightmost column) by QRDNN is the 99% PINP, but even in this case, it is very close (PICP=98.7%). On average, QRDNN intervals are still generally narrower.

We continue with the solar energy regions, starting with the Ciudad Real data (Table 6). As can be seen, the desired coverage for any target PINP cannot be achieved using LQR and GBQR, although GBQR comes close to achieving coverage for the 99% PINP (98% on average). Similar results are observed for the wind energy prediction, as QRF is the best performing method regarding PICP coverage: PICP coverage is achieved using QRF for every hour at PINP values of 80%, 85%, 90%, and 95%. For the PINP target of 99% at 15:00, the coverage is close to but does not meet the desired coverage (98.90%) when using QRF. However, on average (rightmost column), coverage is met using this method. When using NGB, the target coverage is achieved on average for the PINP at 80%, 85%, and 90%, whereas when using SVQR, target coverage is achieved for the PINP at 99%. Lastly, the coverage at all hours is met using QRDNN, and on average, coverage is met for the 80%, 85% and 90% targets. Furthermore, although the coverage is not satisfactory for the 95% and 99% PIs using QRDNN, it is fair to say that it is not far away on average (94.9% and 98%, respectively), and the width is generally lower compared to the rest of the nonlinear methods.

Finally, results for Córdoba (solar energy) are presented in Table 7. Once again, accurate coverage is not achieved using LQR. However, Córdoba is the first region where reasonable coverage for some hours is achieved using GBQR. For example, using this method, coverage is achieved at 09:00 for both the 80%, 85% and 90% PIs, but coverage is not achieved for the rest of the hours. For the 95% and 99% PIs, the PINP for the first 2 and 3 time horizons, respectively, are achieved using GNQR, and coverage is also achieved on average for the 95.5% and 99.5% PIs. As expected, the target coverage at most of the hours for every PI is achieved using QRF, and it is always achieved on average for the other PIs. The behavior of QRDNN is similar to those in the rest of the regions: the coverage is achieved for every hour at the 80% and 85% PIs. For the 90% and 95% PIs, coverage is only not met at 09:00 (the case at 90% PI is close). For the 99% PI, the desired coverage is only reached at 12:00, although it stays quite close to the PINP in the remaining cases. Good performance is achieved for NGB regarding the coverage target (it is achieved for the mean PINP values at 80%, 85% and 90%) while PIs are kept narrow. In addition, almost all PINP values are met on average (except 80%) for SVQR with a PI of similar width to those obtained by QRDNN. We can say that this is the only region where other nonlinear methods (SVQR and NGB) compete with QRDNN regarding PI width (and coverage). However, we will see in the following section how the results can be improved by calibrating the PIs.

In summary, according to the previous analyses, two methods stand out overall, QRF and QRDNN. Using QRF, the target coverage is always achieved, and using QRDNN, the coverage is either achieved or close to being achieved,

Table 4 PICP and AIW results (the latter is shown in parentheses) on the Granada data (wind energy) based on the time horizon for all methods

Method	00:00	03:00	06:00	09:00	12:00	15:00	18:00	21:00	Mean
PINP 80% Prediction Interval									
LQR	0.729	0.780	0.699	0.728	**0.816**	0.715	0.712	0.712	0.737
	(0.156)	(0.167)	(0.192)	(0.209)	(0.280)	(0.186)	(0.169)	(0.189)	(0.194)
GBQR	0.696	0.698	0.633	0.624	0.719	0.674	0.696	0.627	0.671
	(0.129)	(0.142)	(0.163)	(0.161)	(0.196)	(0.132)	(0.134)	(0.157)	(0.152)
NGB	0.740	0,734	0,715	0,659	0,775	0,789	0,786	0,715	0.739
	(0.132)	(0.143)	(0.166)	(0.155)	(0.199)	(0.147)	(0.141)	(0.161)	(0.156)
SVQR	0.742	0.734	0.704	0.714	**0.827**	0.775	0.775	0.734	0.751
	(0.131)	(0.138)	(0.163)	(0.165)	(0.207)	(0.143)	(0.139)	(0.159)	(0.156)
QRF	**0.836**	**0.857**	**0.803**	0.758	**0.841**	**0.874**	**0.885**	**0.841**	**0.837**
	(0.166)	(0.181)	(0.206)	(0.197)	(0.252)	(0.184)	(0.178)	(0.199)	(0.195)
QRDNN	**0.819**	**0.813**	0.795	0.756	**0.874**	**0.825**	**0.838**	0.792	**0.814**
	(0.138)	(0.144)	(0.166)	(0.163)	(0.208)	(0.147)	(0.141)	(0.165)	(0.159)
PINP 85% Prediction Interval									
LQR	0.770	0.821	0.767	0.783	**0.860**	0.827	0.789	0.775	0.799
	(0.175)	(0.188)	(0.216)	(0.233)	(0.315)	(0.217)	(0.195)	(0.216)	(0.219)
GBQR	0.797	0.791	0.723	0.698	0.789	0.778	0.770	0.734	0.760
	(0.152)	(0.166)	(0.192)	(0.187)	(0.229)	(0.157)	(0.160)	(0.183)	(0.178)
NGB	0.792	0.797	0.762	0.706	0.819	0.822	0.847	0.778	0.790
	(0.148)	(0.161)	(0.187)	(0.174)	(0.224)	(0.165)	(0.159)	(0.181)	(0.175)
SVQR	0.808	0.808	0.797	0.755	0.847	0.786	0.819	0.770	0.799
	(0.152)	(0.160)	(0.187)	(0.187)	(0.233)	(0.158)	(0.157)	(0.183)	(0.177)
QRF	**0.890**	**0.896**	**0.858**	0.808	**0.888**	**0.921**	**0.918**	**0.890**	**0.884**
	(0.189)	(0.207)	(0.234)	(0.223)	(0.286)	(0.210)	(0.203)	(0.226)	(0.222)
QRDNN	**0.855**	0.838	0.844	0.816	**0.921**	**0.888**	**0.882**	0.838	**0.860**
	(0.156)	(0.162)	(0.187)	(0.185)	(0.236)	(0.168)	(0.161)	(0.186)	(0.180)
PINP 90% Prediction Interval									
LQR	0.852	0.871	0.819	0.841	**0.901**	0.849	0.855	0.844	0.854
	(0.206)	(0.220)	(0.250)	(0.267)	(0.363)	(0.250)	(0.229)	(0.255)	(0.255)
GBQR	0.855	0.857	0.811	0.750	0.844	0.838	0.858	0.844	0.832
	(0.182)	(0.197)	(0.227)	(0.219)	(0.269)	(0.190)	(0.192)	(0.220)	(0.212)
NGB	0.836	0.852	0.827	0.761	0.866	0.890	0.890	0.849	0.846
	(0.169)	(0.184)	(0.213)	(0.199)	(0.256)	(0.188)	(0.181)	(0.207)	(0.200)
SVQR	0.858	0.865	0.855	0.821	**0.923**	0.896	0.877	0.838	0.867
	(0.179)	(0.185)	(0.216)	(0.219)	(0.283)	(0.198)	(0.190)	(0.216)	(0.211)
QRF	**0.910**	**0.937**	**0.921**	0.863	**0.918**	**0.953**	**0.951**	**0.926**	**0.922**
	(0.221)	(0.241)	(0.273)	(0.259)	(0.333)	(0.242)	(0.237)	(0.261)	(0.258)
QRDNN	0.880	0.882	0.899	0.843	**0.940**	**0.915**	**0.918**	0.880	0.894
	(0.176)	(0.182)	(0.210)	(0.207)	(0.265)	(0.189)	(0.183)	(0.209)	(0.203)

Table 4 (continued)

Method	00:00	03:00	06:00	09:00	12:00	15:00	18:00	21:00	Mean
PINP 95% Prediction Interval									
LQR	0.901	0.920	0.896	0.885	0.943	0.915	0.904	0.915	0.910
	(0.258)	(0.279)	(0.313)	(0.324)	(0.442)	(0.318)	(0.296)	(0.317)	(0.318)
GBQR	0.896	0.920	0.874	0.827	0.888	0.893	0.923	0.912	0.892
	(0.223)	(0.238)	(0.271)	(0.259)	(0.323)	(0.232)	(0.231)	(0.266)	(0.255)
NGB	0.893	0.915	0.888	0.832	0.910	0.932	0.940	0.910	0.902
	(0.202)	(0.219)	(0.254)	(0.237)	(0.305)	(0.224)	(0.216)	(0.246)	(0.238)
SVQR	0.915	**0.951**	0.923	0.901	0.945	0.942	**0.951**	0.918	0.931
	(0.224)	(0.238)	(0.275)	(0.271)	(0.353)	(0.252)	(0.243)	(0.272)	(0.266)
QRF	0.948	**0.975**	**0.959**	0.931	**0.962**	**0.967**	**0.984**	**0.970**	**0.962**
	(0.274)	(0.301)	(0.341)	(0.320)	(0.407)	(0.296)	(0.294)	(0.323)	(0.320)
QRDNN	0.923	0.940	0.934	0.931	**0.973**	0.948	**0.956**	0.937	0.943
	(0.211)	(0.219)	(0.252)	(0.247)	(0.318)	(0.230)	(0.220)	(0.251)	(0.243)
PINP 99% Prediction Interval									
LQR	0.959	0.964	0.943	0.956	0.975	0.948	0.959	0.953	0.957
	(0.349)	(0.373)	(0.417)	(0.415)	(0.576)	(0.423)	(0.416)	(0.441)	(0.426)
GBQR	0.973	0.978	0.973	0.953	0.970	0.975	0.986	0.975	0.973
	(0.323)	(0.350)	(0.404)	(0.377)	(0.490)	(0.371)	(0.351)	(0.391)	(0.3820)
NGB	0.932	0.973	0.948	0.940	0.962	0.978	0.978	0.962	0.959
	(0.265)	(0.288)	(0.334)	(0.311)	(0.401)	(0.295)	(0.284)	(0.323)	(0.313)
SVQR	0.989	**0.992**	**0.992**	0.978	0.984	0.986	0.978	0.984	0.985
	(0.383)	(0.414)	(0.475)	(0.418)	(0.571)	(0.441)	(0.415)	(0.461)	(0.447)
QRF	**0.992**	**0.995**	**0.995**	0.981	**1**	**0.995**	**0.997**	**0.992**	**0.993**
	(0.393)	(0.424)	(0.482)	(0.448)	(0.574)	(0.421)	(0.423)	(0.459)	(0.453)
QRDNN	0.984	0.986	0.975	0.981	**0.992**	**0.995**	**0.992**	0.989	0.987
	(0.288)	(0.300)	(0.344)	(0.337)	(0.445)	(0.331)	(0.315)	(0.348)	(0.339)

This table is divided into 5 subtables, one per PINP target value. The rightmost column of each subtable is the average of the PICP and AIW values across all time horizons. Values in bold indicate that the target PINP is achieved using the method. In terms of the mean, only when using QRDNN and QRF can the target PINP (or close to it) be obtained, but the narrowest intervals (smallest AIW) are obtained using QRDNN

while generally narrower PIs are obtained. From Table 8, the differences between these methods are further explored. For every region in Table 8, there are two rows. The first row presents information on whether QRDNN can be used to obtain the desired coverage (-) and how far off it is (in percentage): min $\left(0, \frac{PICP-PINP}{PINP}\right)\%$. The second row shows (between brackets) the decrease of AIW for QRDNN vs. QRF: $\frac{AIW_{QRF}-AIW_{QRDNN}}{AIW_{QRF}}\%$. Only the average results across all time horizons are considered. It can be seen that the desired PINP is achieved using QRDNN in most cases, and even when coverage is not attained, the difference is smaller than 1.02% in the worst case, which is much smaller in general. However, the intervals using QRDNN are 8% to 29% narrower than those using QRF (8% to 24% if only intervals where $PICP \geq PINP$ are considered).

To complete the PI coverage and width analysis, an example of the 95% PINP for July 2018 using the Lugo data is provided in Fig. 6. The red area represents the interval generated by QRDNN, and the blue area represents the interval generated by QRF.

The real wind power production data is represented by black points.

It is easily noted that the QRF intervals are wider for the majority of the points in the set.

Table 5 Results on the Lugo data (wind energy), which are similar to those in Table 4

Method	00:00	03:00	06:00	09:00	12:00	15:00	18:00	21:00	Mean
PINP 80% Prediction Interval									
LQR	0.756	0.737	0.781	0.786	0.729	0.792	0.740	0.729	0.756
	(0.223)	(0.175)	(0.207)	(0.193)	(0.239)	(0.234)	(0.261)	(0.189)	(0.215)
GBQR	0.698	0.660	0.715	0.715	0.740	0.693	0.696	0.701	0.702
	(0.161)	(0.129)	(0.150)	(0.124)	(0.159)	(0.143)	(0.154)	(0.125)	(0.143)
NGB	**0.867**	**0.825**	**0.838**	**0.836**	**0.808**	**0.814**	**0.816**	**0.825**	**0.829**
	(0.180)	(0.143)	(0.170)	(0.135)	(0.169)	(0.161)	(0.179)	(0.141)	(0.160)
SVQR	0.734	0.759	0.745	0.775	0.775	0.729	0.745	0.767	0.754
	(0.175)	(0.136)	(0.159)	(0.133)	(0.165)	(0.147)	(0.171)	(0.137)	(0.153)
QRF	**0.839**	**0.841**	**0.849**	**0.830**	**0.863**	**0.844**	**0.847**	**0.844**	**0.845**
	(0.197)	(0.155)	(0.179)	(0.147)	(0.187)	(0.171)	(0.191)	(0.157)	(0.173)
QRDNN	**0.825**	**0.803**	**0.822**	**0.811**	**0.849**	**0.852**	**0.833**	**0.841**	**0.827**
	(0.173)	(0.134)	(0.154)	(0.135)	(0.172)	(0.161)	(0.178)	(0.136)	(0.155)
PINP 85% Prediction Interval									
LQR	0.826	0.797	0.836	0.827	0.803	0.874	0.816	0.808	0.823
	(0.265)	(0.205)	(0.244)	(0.221)	(0.277)	(0.279)	(0.315)	(0.228)	(0.254)
GBQR	0.784	0.745	0.795	0.803	0.811	0.778	0.759	0.767	0.780
	(0.188)	(0.152)	(0.177)	(0.146)	(0.184)	(0.166)	(0.178)	(0.145)	(0.167)
NGB	**0.900**	**0.874**	**0.874**	**0.868**	0.847	**0.852**	**0.855**	**0.868**	**0.867**
	(0.203)	(0.161)	(0.191)	(0.151)	(0.190)	(0.181)	(0.201)	(0.158)	(0.179)
SVQR	0.825	0.803	0.803	0.811	0.830	0.803	0.814	0.814	0.813
	(0.197)	(0.154)	(0.179)	(0.151)	(0.189)	(0.172)	(0.195)	(0.154)	(0.174)
QRF	**0.864**	**0.882**	**0.896**	**0.866**	**0.904**	**0.890**	**0.907**	**0.888**	**0.887**
	(0.224)	(0.176)	(0.205)	(0.168)	(0.213)	(0.195)	(0.218)	(0.178)	(0.197)
QRDNN	**0.878**	**0.855**	**0.858**	**0.863**	**0.888**	**0.871**	**0.877**	**0.877**	**0.877**
	(0.193)	(0.145)	(0.172)	(0.150)	(0.192)	(0.180)	(0.199)	(0.152)	(0.174)
PINP 90% Prediction Interval									
LQR	0.881	0.847	0.880	0.871	0.849	**0.912**	0.858	0.866	0.870
	(0.310)	(0.242)	(0.291)	(0.261)	(0.326)	(0.325)	(0.369)	(0.268)	(0.299)
GBQR	0.853	0.825	0.863	0.866	0.858	0.841	0.830	0.838	0.847
	(0.220)	(0.177)	(0.207)	(0.169)	(0.217)	(0.197)	(0.210)	(0.173)	(0.196))
LQR	0.881	0.847	0.880	0.871	0.849	**0.912**	0.858	0.866	0.870
	(0.310)	(0.242)	(0.291)	(0.261)	(0.326)	(0.325)	(0.369)	(0.268)	(0.299)
GBQR	0.853	0.825	0.863	0.866	0.858	0.841	0.830	0.838	0.847
	(0.220)	(0.177)	(0.207)	(0.169)	(0.217)	(0.197)	(0.210)	(0.173)	(0.196))
NGB	**0.942**	**0.910**	**0.929**	**0.915**	0.896	0.885	**0.907**	**0.904**	**0.911**
	(0.232)	(0.184)	(0.218)	(0.173)	(0.217)	(0.207)	(0.230)	(0.181)	(0.205)
SVQR	0.873	0.836	0.882	0.888	0.871	0.868	0.847	0.847	0.864
	(0.234)	(0.184)	(0.213)	(0.176)	(0.214)	(0.193)	(0.219)	(0.180)	(0.202)

Table 5 (continued)

Method	00:00	03:00	06:00	09:00	12:00	15:00	18:00	21:00	Mean
QRF	**0.920**	**0.921**	**0.934**	**0.934**	**0.937**	**0.932**	**0.943**	**0.937**	**0.932**
	(0.263)	(0.207)	(0.239)	(0.196)	(0.250)	(0.227)	(0.254)	(0.208)	(0.230)
QRDNN	**0.906**	0.882	0.899	**0.904**	**0.910**	**0.912**	**0.918**	**0.926**	**0.907**
	(0.222)	(0.172)	(0.197)	(0.172)	(0.221)	(0.207)	(0.223)	(0.175)	(0.200)
PINP 95% Prediction Interval									
LQR	0.936	0.912	0.934	0.932	0.926	0.945	0.912	0.910	0.926
	(0.417)	(0.328)	(0.390)	(0.341)	(0.414)	(0.414)	(0.485)	(0.354)	(0.393)
GBQR	0.911	0.921	0.926	0.932	0.943	0.918	0.918	0.921	0.924
	(0.278)	(0.223)	(0.263)	(0.209)	(0.266)	(0.250)	(0.273)	(0.220)	(0.248)
NGB	**0.972**	**0.953**	**0.964**	0.942	0.942	0.915	0.926	**0.953**	0.946
	(0.276)	(0.219)	(0.260)	(0.206)	(0.258)	(0.246)	(0.274)	(0.215)	(0.244)
SVQR	0.934	0.910	0.940	0.940	0.926	0.929	0.942	0.923	0.930
	(0.303)	(0.240)	(0.278)	(0.227)	(0.275)	(0.256)	(0.292)	(0.238)	(0.264)
QRF	**0.975**	**0.967**	**0.970**	**0.975**	**0.973**	**0.973**	**0.967**	**0.970**	**0.971**
	(0.326)	(0.256)	(0.300)	(0.243)	(0.310)	(0.281)	(0.314)	(0.259)	(0.286)
QRDNN	**0.953**	0.936	0.936	**0.953**	**0.956**	**0.956**	**0.962**	**0.951**	**0.950**
	(0.265)	(0.206)	(0.236)	(0.205)	(0.263)	(0.247)	(0.274)	(0.209)	(0.238)
NGB	**0.942**	**0.910**	**0.929**	**0.915**	0.896	0.885	**0.907**	**0.904**	**0.911**
	(0.232)	(0.184)	(0.218)	(0.173)	(0.217)	(0.207)	(0.230)	(0.181)	(0.205)
SVQR	0.873	0.836	0.882	0.888	0.871	0.868	0.847	0.847	0.864
	(0.234)	(0.184)	(0.213)	(0.176)	(0.214)	(0.193)	(0.219)	(0.180)	(0.202)
QRF	**0.920**	**0.921**	**0.934**	**0.934**	**0.937**	**0.932**	**0.943**	**0.937**	**0.932**
	(0.263)	(0.207)	(0.239)	(0.196)	(0.250)	(0.227)	(0.254)	(0.208)	(0.230)
QRDNN	**0.906**	0.882	0.899	**0.904**	**0.910**	**0.912**	**0.918**	**0.926**	**0.907**
	(0.222)	(0.172)	(0.197)	(0.172)	(0.221)	(0.207)	(0.223)	(0.175)	(0.200)
PINP 95% Prediction Interval									
LQR	0.936	0.912	0.934	0.932	0.926	0.945	0.912	0.910	0.926
	(0.417)	(0.328)	(0.390)	(0.341)	(0.414)	(0.414)	(0.485)	(0.354)	(0.393)
GBQR	0.911	0.921	0.926	0.932	0.943	0.918	0.918	0.921	0.924
	(0.278)	(0.223)	(0.263)	(0.209)	(0.266)	(0.250)	(0.273)	(0.220)	(0.248)
NGB	**0.972**	**0.953**	**0.964**	0.942	0.942	0.915	0.926	**0.953**	0.946
	(0.276)	(0.219)	(0.260)	(0.206)	(0.258)	(0.246)	(0.274)	(0.215)	(0.244)
SVQR	0.934	0.910	0.940	0.940	0.926	0.929	0.942	0.923	0.930
	(0.303)	(0.240)	(0.278)	(0.227)	(0.275)	(0.256)	(0.292)	(0.238)	(0.264)
QRF	**0.975**	**0.967**	**0.970**	**0.975**	**0.973**	**0.973**	**0.967**	**0.970**	**0.971**
	(0.326)	(0.256)	(0.300)	(0.243)	(0.310)	(0.281)	(0.314)	(0.259)	(0.286)
QRDNN	**0.953**	0.936	0.936	**0.953**	**0.956**	**0.956**	**0.962**	**0.951**	**0.950**
	(0.265)	(0.206)	(0.236)	(0.205)	(0.263)	(0.247)	(0.274)	(0.209)	(0.238)
PINP 99% Prediction Interval									
LQR	0.953	0.959	0.981	0.975	0.967	0.981	0.970	0.934	0.965
	(0.585)	(0.459)	(0.544)	(0.445)	(0.568)	(0.591)	(0.683)	(0.494)	(0.546)

Table 5 (continued)

Method	00:00	03:00	06:00	09:00	12:00	15:00	18:00	21:00	Mean
GBQR	0.986 (0.452)	0.989 (0.363)	**0.992** (0.422)	0.984 (0.336)	0.981 (0.429)	0.975 (0.401)	0.986 (0.440)	0.986 (0.359)	0.985 (0.400)
NGB	0.989 (0.363)	0.981 (0.287)	0.984 (0.341)	0.981 (0.271)	0.975 (0.339)	0.964 (0.323)	0.975 (0.360)	0.981 (0.283)	0.979 (0.321)
SVQR	**0.992** (0.562)	0.989 (0.435)	0.989 (0.505)	0.986 (0.388)	**0.992** (0.497)	**0.997** (0.513)	**0.995** (0.603)	**0.997** (0.464)	**0.992** (0.496)
QRF	**0.995** (0.466)	**0.992** (0.363)	0.989 (0.431)	**0.995** (0.342)	**0.997** (0.437)	**0.995** (0.395)	**1** (0.447)	**0.997** (0.370)	**0.995** (0.406)
QRDNN	0.981 (0.364)	0.975 (0.282)	0.984 (0.323)	0.986 (0.281)	**0.997** (0.362)	**0.995** (0.343)	**0.995** (0.380)	0.986 (0.288)	0.987 (0.328)

In terms of the mean PICP (rightmost column), QRF, QRDNN, and NGB can be utilized to achieve the required coverage or are close to it, but the intervals generated using QRDNN are generally narrower

Table 6 Results on the Ciudad Real data (solar energy), which are similar to those in Table 4

Method	09:00	12:00	15:00	Mean
PINP 80% Prediction Interval				
LQR	0.529 (0.167)	0.471 (0.178)	0.512 (0.193)	0.504 (0.179)
GBQR	0.726 (0.232)	0.775 (0.204)	0.677 (0.165)	0.725 (0.200)
NGB	0.792 (0.277)	**0.866** (0.245)	**0.847** (0.195)	**0.835** (0.239)
SVQR	0.764 (0.225)	0.718 (0.202)	0.699 (0.182)	0.727 (0.203)
QRF	**0.882** (0.310)	**0.912** (0.298)	**0.849** (0.245)	**0.879** (0.284)
QRDNN	**0.830** (0.227)	**0.874** (0.234)	**0.849** (0.214)	**0.821** (0.225)
PINP 85% Prediction Interval				
LQR	0.606 (0.199)	0.564 (0.213)	0.595 (0.226)	0.588 (0.217)
GBQR	0.825 (0.286)	0.844 (0.246)	0.789 (0.196)	0.819 (0.243)
NGB	0.836 (0.311)	**0.893** (0.275)	**0.868** (0.219)	**0.866** (0.268)
SVQR	0.816 (0.264)	0.816 (0.234)	0.762 (0.208)	0.798 (0.235)
QRF	**0.921** (0.362)	**0.937** (0.339)	**0.885** (0.279)	**0.914** (0.327)
QRDNN	**0.874** (0.261)	**0.918** (0.266)	**0.888** (0.242)	**0.893** (0.256)

Table 6 (continued)

Method	09:00	12:00	15:00	Mean
PINP 90% Prediction Interval				
LQR	0.682	0.671	0.669	0.674
	(0.236)	(0.253)	(0.261)	(0.250)
GBQR	0.863	0.890	0.869	0.874
	(0.371)	(0.298)	(0.231)	(0.300)
NGB	0.874	**0.937**	0.899	**0.903**
	(0.355)	(0.314)	(0.250)	(0.307)
SVQR	0.858	0.858	0.822	0.846
	(0.313)	(0.268)	(0.242)	(0.274)
QRF	**0.956**	**0.959**	**0.940**	**0.952**
	(0.438)	(0.397)	(0.325)	(0.387)
QRDNN	**0.904**	**0.945**	**0.904**	**0.918**
	(0.301)	(0.306)	(0.277)	(0.295)
PINP 95% Prediction Interval				
LQR	0.740	0.751	0.753	0.748
	(0.277)	(0.302)	(0.303)	(0.294)
GBQR	0.929	0.932	0.926	0.929
	(0.464)	(0.364)	(0.281)	(0.370)
NGB	0.932	**0.964**	0.921	0.939
	(0.423)	(0.374)	(0.298)	(0.365)
SVQR	0.932	0.918	0.874	0,908
	(0.400)	(0.330)	(0.294)	(0.349)
QRF	**0.978**	**0.984**	**0.970**	**0.977**
	(0.568)	(0.495)	(0.399)	(0.488)
QRDNN	0.948	**0.964**	0.934	0.949
	(0.361)	(0.362)	(0.324)	(0.349)
PINP 99% Prediction Interval				
LQR	0.784	0.792	0.803	0.793
	(0.298)	(0.329)	(0.325)	(0.317)
GBQR	0.986	0.989	0.964	0.980
	(0.788)	(0.579)	(0.448)	(0.605)
NGB	0.984	0.984	0.951	0.973
	(0.556)	(0.492)	(0.392)	(0.480)
SVQR	0.986	**0.992**	**0.992**	**0.990**
	(0.789)	(0.597)	(0.490)	(0.625)
QRF	**1**	**0.997**	0.989	**0.995**
	(0.821)	(0.688)	(0.546)	(0.685)
QRDNN	0.978	0.981	0.981	0.980
	(0.508)	(0.504)	(0.443)	(0.485)

In terms of the mean PICP (rightmost column), QRF, QRDNN, and NGB can be used to achieve the required coverage or close to it, but QRDNN PIs are generally narrower

Table 7 Results on the Córdoba data (solar energy), which are similar to those in Table 4

Method	09:00	12:00	15:00	Mean
PINP 80% Prediction Interval				
LQR	0.696	0.674	0.627	0.666
	(0.136)	(0.183)	(0.148)	(0.156)
GBQR	**0.827**	0.696	0.641	0.722
	(0.180)	(0.174)	(0.118)	(0.157)
NGB	0.786	**0.849**	**0.836**	**0.824**
	(0.188)	(0.198)	(0.128)	(0.171)
SVQR	0.792	0.775	0.745	0.771
	(0.163)	(0.182)	(0.135)	(0.160)
QRF	**0.885**	**0.869**	**0.800**	**0.851**
	(0.217)	(0.250)	(0.168)	(0.212)
QRDNN	**0.836**	**0.874**	**0.808**	**0.839**
	(0.252)	(0.188)	(0.143)	(0.194)
PINP 85% Prediction Interval				
LQR	0.773	0.795	0.729	0.765
	(0.166)	(0.229)	(0.182)	(0.192)
GBQR	**0.880**	0.786	0.748	0.805
	(0.225)	(0.212)	(0.144)	(0.193)
NGB	0.838	**0.901**	**0.868**	**0.869**
	(0.211)	(0.222)	(0.144)	(0.193)
SVQR	**0.855**	**0.877**	0.836	**0.856**
	(0.198)	(0.221)	(0.162)	(0.193)
QRF	**0.923**	**0.896**	0.838	**0.886**
	(0.256)	(0.289)	(0.197)	(0.248)
QRDNN	**0.866**	**0.934**	**0.880**	**0.893**
	(0.286)	(0.215)	(0.161)	(0.248)
PINP 90% Prediction Interval				
LQR	0.841	0.838	0.803	0.827
	(0.198)	(0.274)	(0.216)	(0.229)
GBQR	**0.943**	0.877	0.852	0.890
	(0.303)	(0.261)	(0.176)	(0.247)
NGB	0.879	**0.942**	**0.904**	**0.909**
	(0.241)	(0.254)	(0.165)	(0.220)
SVQR	0.899	**0.923**	**0.915**	**0.912**
	(0.240)	(0.266)	(0.194)	(0.234)
QRF	**0.953**	**0.940**	**0.915**	**0.936**
	(0.317)	(0.349)	(0.234)	(0.300)
QRDNN	0.899	**0.964**	**0.915**	**0.926**
	(0.333)	(0.255)	(0.189)	(0.259)
PINP 95% Prediction Interval				
LQR	0.882	0.860	0.858	0.867
	(0.231)	(0.317)	(0.251)	(0.267)
GBQR	**0.975**	**0.951**	0.940	**0.955**
	(0.428)	(0.337)	(0.220)	(0.328)
NGB	0.934	**0.975**	0.937	0.949
	(0.287)	(0.303)	(0.196)	(0.262)
SVQR	**0.956**	**0.973**	**0.964**	**0.964**
	(0.334)	(0.352)	(0.260)	(0.315)
QRF	**0.978**	**0.975**	**0.984**	**0.979**
	(0.417)	(0.452)	(0.299)	(0.389)
QRDNN	0.943	**0.981**	**0.951**	**0.958**
	(0.400)	(0.306)	(0.223)	(0.310)
PINP 99% Prediction Interval				
LQR	0.915	0.890	0.877	0.894
	(0.247)	(0.341)	(0.267)	(0.285)
GBQR	**0.997**	**0.995**	**0.995**	**0.995**
	(0.836)	(0.708)	(0.439)	(0.661)
NGB	0.970	**0.995**	0.978	0.981
	(0.378)	(0.398)	(0.258)	(0.344)
SVQR	**0.995**	**0.997**	**0.995**	**0.995**
	(0.637)	(0.617	(0.411)	(0.555)
QRF	**1**	**0.997**	**0.995**	**0.997**
	(0.648)	(0.670)	(0.429)	(0.582)
QRDNN	0.984	**0.995**	0.989	0.989
	(0.563)	(0.437)	(0.307)	(0.436)

Using QRF, QRDNN, SVQR, and NGB, the target coverage is reached or is close to being reached. Here, NGB and SVQR compete with QRDNN in terms of narrow PIs

The PICP is a crisp metric in the sense that for an individual instance, its value is either 0 or 1. However, if an instance is outside the PI but very close to the PI bound, the PICP value will still be 0. A smoother understanding of the obtained PIs is presented using WS ((29)). Its value is basically the interval width plus a penalization value, which is linear with the distance between the instance and the PI bounds (the penalization value is zero if the instance is within the PI). Thus, if the instance is outside the PI, but not too far away from the lower or upper bounds, the penalization will be low. However, the penalization value grows quickly with distance, as it is weighted by $\frac{2}{1-PINP}$).

Table 8 Comparison between QRDNN and QRF

Region	80%	85%	90%	95%	99%
Granada (wind)	-	-	-0.62%	-0.76%	-0.34%
	19%	19%	22%	24%	25%
Lugo (wind)	-	-	-	-	-0.27%
	10%	12%	13%	17%	19%
Ciudad Real (solar)	-	-	-	-0.12%	-1.02%
	21%	22%	24%	28%	29%
Córdoba (solar)	-	-	-	-	-0.10 %
	8%	11%	14%	20%	25%

For every region, there are two rows. First row: (-) means the target coverage is achieved, otherwise, the difference (percentage) between PICP and PINP for QRDNN are provided. Second row: decrease (percentage) in AIW of QRDNN vs. QRF. The PINP is achieved in all cases for QRDNN, except for high PINP values, which deviate no further than 1.02%, while the AIW increases from 8% to 29%

In Table 9, the mean value of the Winkler score for each region, method and PI are shown. The best WS value is shown in bold. In the first wind region (Granada), the lowest WS for all coverage is achieved using QRDNN, except for WS99, where the best coverage is achieved using QRF. For the Lugo data, the best score for every coverage using QRDNN, except for WS90, where the performance of QRF is slightly better. These results coincide with those in which the target PINP is achieved or almost met for a given method. In general, the worst values are obtained using LQR and GBQR.

In addition, in the solar energy regions (Ciudad Real and Córdoba), we find that QRDNN is the best performing method for every coverage and region, except for the PINP at 99% for the Ciudad Real data, where SVQR performs slightly better.

In summary, QRDNN is the best performer for the WS metric, except for a few cases.

We conclude this section of the analysis by commenting the final metric: the (mean) coverage-width ratio (Table 10). First, we consider the fact that narrow intervals can be achieved at the cost of large differences with respect to the required coverage for some methods. Methods whose coverage deviates from the target PINP by more than one unit have been represented using a smaller font, and they are not taken into account when computing the best ratio. Thus, for example, for the PINP value of 99%, we only take into account methods with a PICP value equal or greater than 98% to compute the best ratio (in bold).

It can be seen that QRDNN is clearly the best performing method for both the wind energy regions (Granada and Lugo), and for one solar energy region (Ciudad Real). This DNN-based method is only surpassed on the Córdoba data by NGB. As we will see in the next section, this may be caused by a bad quantile calibration, where the constructed PIs may be wider than necessary.

5.3 Prediction interval estimation with calibrated quantiles

Given the results regarding quantile and PI estimation and taking coverage, width and ratio into account, QRF and QRDNN are considered to be the two best performing methods in the regions of Granada, Lugo, and Ciudad Real, where both methods have achieved robust performance. However, in the region of Córdoba, NGB is considered

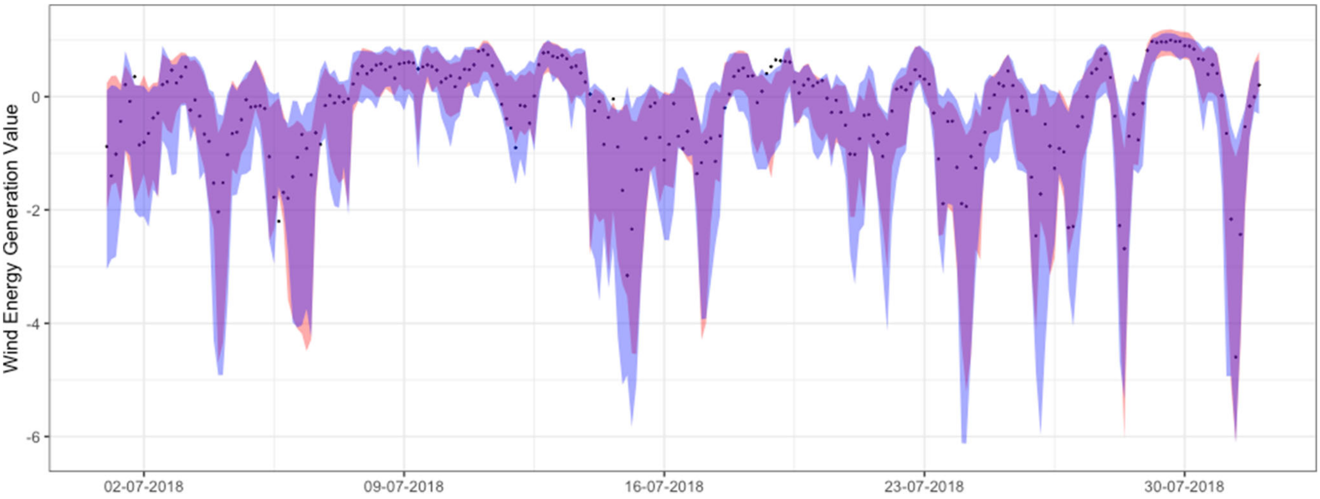


Fig. 6 Example of the 95% prediction interval using QRDNN (red area) and QRF (blue area) for the Lugo data (July 2018). The real wind energy production data is represented by black points. The QRDNN provides narrower intervals for the majority of the points

Table 9 Mean Winkler score by region and method for every PINP target value

Region	Method	WS80	WS85	WS90	WS95	WS99
Granada (wind)	LQR	2.05	2.24	2.55	3.23	5.77
	GBQR	1.79	1.93	2.18	2.70	4.31
	NGB	1.77	1.96	2.25	2.83	5.25
	SVQR	1.67	1.83	2.06	2.49	4.00
	QRF	1.68	1.85	2.07	2.45	**3.48**
	QRDNN	**1.54**	**1.69**	**1.90**	**2.29**	3.62
Lugo (wind)	LQR	1.64	1.77	1.99	2.43	3.53
	GBQR	1.12	1.21	1.35	1.58	2.42
	NGB	1.14	1.24	1.39	1.65	2.33
	SVQR	1.13	1.22	1.37	1.64	2.82
	QRF	1.07	**1.15**	**1.27**	1.50	2.02
	QRDNN	**1.06**	**1.15**	1.28	**1.48**	**1.93**
Ciudad Real (solar)	LQR	1.91	2.10	2.40	3.28	9.74
	GBQR	1.32	1.45	1.66	2.04	3.47
	NGB	1.43	1.57	1.78	2.17	3.57
	SVQR	1.36	1.53	1.76	2.16	**2.67**
	QRF	1.36	1.50	1.71	2.09	2.91
	QRDNN	**1.26**	**1.39**	**1.56**	**1.89**	2.87
Córdoba (solar)	LQR	1.31	1.45	1.65	2.05	4.57
	GBQR	1.09	1.21	1.39	1.75	3.06
	NGB	1.12	1.22	1.37	1.63	2.52
	SVQR	1.09	1.19	1.33	1.68	2.57
	QRF	1.13	1.28	1.47	1.83	2.72
	QRDNN	**1.04**	**1.15**	**1.29**	**1.52**	**2.06**

The best value for each region is shown in bold. The best values are achieved using QRDNN in most cases

jointly with QRDNN due to its good results in relation to the coverage-width ratio (Table 10).

In this section, we show how improvements in the PI quality can be made by following the conformalized regression methodology presented in Section 4.4. For this purpose, we report the coverage and width of the above mentioned methods in their conformalized forms.

First, Table 11 shows the PICP and AIW results on the Granada data using the conformalized forms of QRF (CQRF) and QRDNN (CQRDNN). Generally, we can see that conformalizing reduces the coverage. This may be due to the fact that a larger coverage than required is obtained using these methods, and calibration with the validation set reduces it, which also results in narrower PIs. Nevertheless, although in some cases the calibrated PIs do not achieve the target PINP, there is never a large deviation, with the advantage that AIW is reduced. DNN-based methods (QRDNN and CQRDNN) remain the methods with the best performance due to their narrow PIs.

In Table 12, the PICP and AIW results on the Lugo data using CQRF and CQRDNN are shown. Similarly to the case of the Granada data, the coverage tends to be reduced when there an excess of coverage with respect to the target PINP is found in the validation set for the conformalized methods. As a result, the PICP values obtained by CQRF and CQRDNN are closer to the PINP and in some cases below it. However, the improvement in AIW makes conformalizing worthwhile, especially for the PINP at 80%, 85%, and 90%, where the improvement in AIW is exceptional (see the mean column in Table 12). For the 99% PINP, the AIW value is slightly larger, but the coverage is also increased, which is what is required in this case.

Table 13 shows the PI estimation performance of the conformalized methods, CQRF and CQRDNN, for the Ciudad Real data (solar). For CQRF, there is only a slight reduction in coverage from the calibration quantiles result, but this still results in a significant improvement in the PI width, especially for QRDNN and the 80%, 85%, and

Table 10 Mean ratio score (PICP/AIW) by region and method for every target PINP

Region	Method	Ratio 80	Ratio 85	Ratio 90	Ratio 95	Ratio 99
Granada (wind)	LQR	3.90	3.73	3.43	2.92	2.29
	GBQR	4.50	4.34	3.99	3.55	2.58
	NGB	4.82	4.59	4.30	3.85	3.11
	SVQR	4.90	4.59	4.18	3.56	2.23
	QRF	4.35	4.04	3.62	3.05	2.22
	QRDNN	**5.20**	**4.84**	**4.48**	**3.93**	**2.96**
Lugo (wind)	LQR	3.57	3.29	2.96	2.39	1.8
	GBQR	4.96	4.71	4.36	3.77	2.49
	NGB	5.25	4.89	4.49	3.92	3.09
	SVQR	4.99	4.72	4.33	3.56	2.03
	QRF	4.93	4.55	4.09	3.43	2.48
	QRDNN	**5.40**	**5.08**	**4.60**	**4.04**	**3.05**
Ciudad Real (solar)	LQR	2.82	2.78	2.70	2.55	2.50
	GBQR	3.68	3.44	3.02	2.62	1.70
	NGB	3.58	3.30	3.01	2.62	2.06
	SVQR	3.59	3.42	3.11	2.69	1.65
	QRF	3.13	2.83	2.50	2.05	1.49
	QRDNN	**3.79**	**3.49**	**3.12**	**2.73**	**2.03**
Córdoba (solar)	LQR	4.34	4.04	3.68	3.31	3.20
	GBQR	4.68	4.28	3.77	3.12	1.62
	NGB	**4.99**	**4.68**	**4.28**	**3.75**	**2.95**
	SVQR	4.89	4.49	3.97	3.11	1.87
	QRF	4.11	3.65	3.20	2.60	1.78
	QRDNN	4.55	4.28	3.78	3.27	2.42

The best value for each region is shown in bold. Methods whose coverage deviates from the target PINP by more than one unit have been represented using a smaller font. QRDNN is the best method for the Granada, Lugo and Ciudad Real data

90% target PINPs. There is some AIW increase for the 95% and 99% PINPs, but for the 95% case, this result is actually required to increase the coverage. Although both methods benefit from calibration, conformalized QRDNN is still better than CQRF in terms of AIW.

Results for the Córdoba data are displayed in Table 14. As previously mentioned, NGB was chosen to compare with QRDNN in this region. It is interesting to note that calibration does not improve the PIs for NGB (CNGB), as excessive coverage and a larger AIW are obtained. On the other hand, PIs are greatly improved using CQRDNN: coverage is closer to the PINP target, satisfying it, and the AIW decreases. Furthermore, the employment of calibrated quantiles makes CQRDNN the best performing method for this region, and are superior to the original results using NGB.

In summary, in most cases, calibrated quantiles (conformalized quantile regression) result in a PICP value that is closer to the target PINP value and a decreased AIW. In particular, CQRDNN benefits particularly from calibration, as also shown in Table 15, which shows that the coverage-width ratio improves when using CQRDNN instead of QRDNN. As can be seen, improvements occur for most PINP values and regions, especially for Córdoba. We note that it is more difficult to improve the ratio for the PINP at 99%, which is understandable due the high coverage requirements. In general, we can confirm that employing calibrated quantiles improves the PI quality. Overall, results show the good performance of deep NN-based methods.

5.4 Analysis by season

Generally, a better overall performance in relation to prediction interval coverage, width, and quality for the time horizons analyzed is achieved using CQRDNN. To complete this section of results, PIs obtained using this method are studied from a seasonal perspective. Thus, predictions made on the test set will be disaggregated into the four seasons of the year to check for possible variability during the year.

Table 11 PICP and AIW (in parenthesis) results on the Granada data (wind energy) based on the time horizon for QRF and QRDNN and their conformalized forms (CQRF and CQRDNN, respectively)

Method	00:00	03:00	06:00	09:00	12:00	15:00	18:00	21:00	Mean
PINP 80% Prediction Interval									
QRF	**0.836**	**0.857**	**0.803**	0.758	**0.841**	**0.874**	**0.885**	**0.841**	**0.837**
	(0.166)	(0.181)	(0.206)	(0.197)	(0.252)	(0.184)	(0.178)	(0.199)	(0.195)
CQRF	0.792	**0.810**	0.756	0.758	**0.841**	**0.822**	0.781	0.759	0.790
	(0.158)	(0.169)	(0.195)	(0.200)	(0.253)	(0.171)	(0.155)	(0.180)	(0.185)
QRDNN	**0.819**	**0.813**	0.795	0.756	**0.874**	**0.825**	**0.838**	0.792	**0.814**
	(0.138)	(0.144)	(0.166)	(0.163)	(0.208)	(0.147)	(0.141)	(0.165)	(0.159)
CQRDNN	**0.822**	**0.813**	0.764	0.750	**0.836**	0.784	**0.800**	0.773	0.793
	(0.141)	(0.148)	(0.153)	(0.157)	(0.200)	(0.132)	(0.125)	(0.157)	(0.152)
PINP 85% Prediction Interval									
QRF	**0.890**	**0.896**	**0.858**	0.808	**0.888**	**0.921**	**0.918**	**0.890**	**0.884**
	(0.189)	(0.207)	(0.234)	(0.223)	(0.286)	(0.210)	(0.203)	(0.226)	(0.222)
CQRF	0.847	0.846	0.805	0.816	**0.877**	0.847	0.833	0.814	0.836
	(0.179)	(0.193)	(0.219)	(0.225)	(0.283)	(0.191)	(0.180)	(0.204)	(0.209)
QRDNN	**0.855**	0.838	0.844	0.816	**0.921**	**0.888**	**0.882**	0.838	**0.860**
	(0.156)	(0.162)	(0.187)	(0.185)	(0.236)	(0.168)	(0.161)	(0.186)	(0.180)
CQRDNN	**0.852**	**0.863**	0.822	0.810	**0.890**	0.847	**0.852**	0.825	0.845
	(0.159)	(0.170)	(0.176)	(0.181)	(0.226)	(0.147)	(0.140)	(0.177)	(0.172)
PINP 90% Prediction Interval									
QRF	**0.910**	**0.937**	**0.921**	0.863	**0.918**	**0.953**	**0.951**	**0.926**	**0.922**
	(0.221)	(0.241)	(0.273)	(0.259)	(0.333)	(0.242)	(0.237)	(0.261)	(0.258)
CQRF	0.893	**0.901**	0.871	0.854	0.896	**0.912**	0.885	0.866	0.885
	(0.214)	(0.228)	(0.256)	(0.257)	(0.322)	(0.229)	(0.211)	(0.238)	(0.244)

Table 11 (continued)

Method	00:00	03:00	06:00	09:00	12:00	15:00	18:00	21:00	Mean
QRDNN	0.880	0.882	0.899	0.843	**0.940**	**0.915**	**0.918**	0.880	0.894
	(0.176)	(0.182)	(0.210)	(0.207)	(0.265)	(0.189)	(0.183)	(0.209)	(0.203)
CQRDNN	**0.901**	0.896	0.858	0.846	0.929	0.899	0.890	0.868	0.886
	(0.185)	(0.193)	(0.198)	(0.200)	(0.258)	(0.177)	(0.160)	(0.202)	(0.197)
PINP 95% Prediction Interval									
QRF	0.948	**0.975**	**0.959**	0.931	**0.962**	**0.967**	**0.984**	**0.970**	**0.962**
	(0.274)	(0.301)	(0.341)	(0.320)	(0.407)	(0.296)	(0.294)	(0.323)	(0.320)
CQRF	0.945	0.945	0.945	0.907	**0.953**	0.948	**0.956**	0.934	0.942
	(0.273)	(0.284)	(0.327)	(0.312)	(0.403)	(0.283)	(0.271)	(0.304)	(0.307)
QRDNN	0.923	0.940	0.934	0.931	**0.973**	0.948	**0.956**	0.937	0.943
	(0.211)	(0.219)	(0.252)	(0.247)	(0.318)	(0.230)	(0.220)	(0.251)	(0.243)
CQRDNN	0.945	0.942	0.926	0.937	**0.964**	**0.953**	0.942	0.921	0.941
	(0.226)	(0.220)	(0.247)	(0.246)	(0.316)	(0.220)	(0.204)	(0.233)	(0.239)
PINP 99% Prediction Interval									
QRF	**0.992**	**0.995**	**0.995**	0.981	**1**	**0.995**	**0.997**	**0.992**	**0.993**
	(0.393)	(0.424)	(0.482)	(0.448)	(0.574)	(0.421)	(0.423)	(0.459)	(0.453)
CQRF	**0.992**	**0.992**	**0.992**	0.956	0.986	0.986	0.970	**0.992**	0.983
	(0.390)	(0.413)	(0.466)	(0.437)	(0.558)	(0.412)	(0.401)	(0.455)	(0.442)
QRDNN	0.984	0.986	0.975	0.981	**0.992**	**0.995**	**0.992**	0.989	0.987
	(0.288)	(0.300)	(0.344)	(0.337)	(0.445)	(0.331)	(0.315)	(0.348)	(0.339)
CQRDNN	**0.995**	**0.992**	0.978	0.981	0.986	0.989	0.986	**0.992**	0.987
	(0.308)	(0.363)	(0.360)	(0.329)	(0.409)	(0.279)	(0.288)	(0.397)	(0.342)

There is one subtable for each PINP target value. The rightmost column of each subtable is the average of the PICP and AIW results across all time horizons. Values in bold indicate that a PICP value equal to or greater than the target PINP is achieved for a given method. In terms of the mean, the coverage is reduced when using the conformalized methods, but the resulting PICP values are not far from the target PINP, with the advantage that AIW is reduced in most cases (except for the 99% target). CQRDNN is generally the best performer in terms of AIW

Table 12 PICP and AIW (in parenthesis) results on the Lugo data (wind energy) based on the time horizon for QRF and QRDNN and their conformalized forms (CQRF and CQRDNN, respectively)

Method	00:00	03:00	06:00	09:00	12:00	15:00	18:00	21:00	Mean
PINP 80% Prediction Interval									
QRF	**0.839**	**0.841**	**0.849**	**0.830**	**0.863**	**0.844**	**0.847**	**0.844**	**0.845**
	(0.197)	(0.155)	(0.179)	(0.147)	(0.187)	(0.171)	(0.191)	(0.157)	(0.173)
CQRF	**0.820**	**0.816**	0.770	0.781	**0.830**	0.797	0.762	0.762	0.792
	(0.196)	(0.153)	(0.170)	(0.141)	(0.180)	(0.161)	(0.177)	(0.146)	(0.165)
QRDNN	**0.825**	**0.803**	**0.822**	**0.811**	**0.849**	**0.852**	**0.833**	**0.841**	**0.827**
	(0.173)	(0.134)	(0.154)	(0.135)	(0.172)	(0.161)	(0.178)	(0.136)	(0.155)
CQRDNN	**0.817**	**0.800**	0.781	0.789	**0.822**	0.781	**0.827**	0.797	**0.802**
	(0.165)	(0.130)	(0.141)	(0.126)	(0.158)	(0.140)	(0.166)	(0.123)	(0.144)
PINP 85% Prediction Interval									
QRF	**0.864**	**0.882**	**0.896**	**0.866**	**0.904**	**0.890**	**0.907**	**0.888**	**0.887**
	(0.224)	(0.176)	(0.205)	(0.168)	(0.213)	(0.195)	(0.218)	(0.178)	(0.197)
CQRF	**0.850**	**0.866**	0.825	0.833	**0.888**	0.838	0.827	0.803	0.841
	(0.220)	(0.174)	(0.193)	(0.162)	(0.209)	(0.183)	(0.202)	(0.167)	(0.189)
QRDNN	**0.878**	**0.855**	**0.858**	**0.863**	**0.888**	**0.871**	**0.877**	**0.877**	**0.877**
	(0.193)	(0.145)	(0.172)	(0.150)	(0.192)	(0.180)	(0.199)	(0.152)	(0.174)
CQRDNN	**0.870**	0.849	0.819	0.822	**0.855**	0.847	**0.863**	0.836	0.845
	(0.189)	(0.148)	(0.157)	(0.138)	(0.181)	(0.163)	(0.185)	(0.136)	(0.162)
PINP 90% Prediction Interval									
QRF	**0.920**	**0.921**	**0.934**	**0.934**	**0.937**	**0.932**	**0.943**	**0.937**	**0.932**
	(0.263)	(0.207)	(0.239)	(0.196)	(0.250)	(0.227)	(0.254)	(0.208)	(0.230)
CQRF	0.898	**0.915**	0.871	**0.910**	**0.937**	0.896	0.877	0.893	**0.900**
	(0.258)	(0.204)	(0.227)	(0.192)	(0.250)	(0.216)	(0.234)	(0.201)	(0.223)

Table 12 (continued)

Method	00:00	03:00	06:00	09:00	12:00	15:00	18:00	21:00	Mean
QRDNN	**0.906**	0.882	0.899	**0.904**	**0.910**	**0.912**	**0.918**	**0.926**	**0.907**
	(0.222)	(0.172)	(0.197)	(0.172)	(0.221)	(0.207)	(0.223)	(0.175)	(0.200)
CQRDNN	**0.909**	0.893	0.868	0.874	**0.932**	0.888	0.882	**0.910**	0.894
	(0.220)	(0.164)	(0.178)	(0.158)	(0.218)	(0.182)	(0.206)	(0.166)	(0.186)
PINP 95% Prediction Interval									
QRF	**0.975**	**0.967**	**0.970**	**0.975**	**0.973**	**0.973**	**0.967**	**0.970**	**0.971**
	(0.326)	(0.256)	(0.300)	(0.243)	(0.310)	(0.281)	(0.314)	(0.259)	(0.286)
CQRF	**0.958**	**0.956**	0.942	**0.956**	**0.978**	0.948	0.945	0.932	**0.952**
	(0.321)	(0.251)	(0.293)	(0.238)	(0.312)	(0.271)	(0.300)	(0.251)	(0.279)
QRDNN	**0.953**	0.936	0.936	**0.953**	**0.956**	**0.956**	**0.962**	**0.951**	**0.950**
	(0.265)	(0.206)	(0.236)	(0.205)	(0.263)	(0.247)	(0.274)	(0.209)	(0.238)
CQRDNN	**0.961**	0.926	0.937	0.918	**0.975**	0.948	0.945	0.940	0.944
	(0.287)	(0.202)	(0.235)	(0.193)	(0.284)	(0.231)	(0.252)	(0.197)	(0.235)
PINP 99% Prediction Interval									
QRF	**0.995**	**0.992**	0.989	**0.995**	**0.997**	**0.995**	**1**	**0.997**	**0.995**
	(0.466)	(0.363)	(0.431)	(0.342)	(0.437)	(0.395)	(0.447)	(0.370)	(0.406)
CQRF	**0.992**	0.981	0.986	**0.995**	**1**	0.989	**1**	**0.992**	**0.992**
	(0.465)	(0.354)	(0.427)	(0.341)	(0.457)	(0.386)	(0.499)	(0.363)	(0.411)
QRDNN	0.981	0.975	0.984	0.986	**0.997**	**0.995**	**0.995**	0.986	0.987
	(0.364)	(0.282)	(0.323)	(0.281)	(0.362)	(0.343)	(0.380)	(0.288)	(0.328)
CQRDNN	0.989	0.984	0.981	0.984	**0.997**	**0.992**	0.989	**1**	0.989
	(0.367)	(0.332)	(0.314)	(0.284)	(0.434)	(0.323)	(0.343)	(0.417)	(0.352)

In terms of the mean, the coverage is reduced when using the conformalized methods, but the resulting PICP values are not far from the target PINP, with the advantage that AIW is reduced in most cases. CQRDNN is generally the best performer in terms of AIW

Table 13 PICP and AIW (in parenthesis) results on the Ciudad Real data (solar energy) based on the time horizon for QRF and QRDNN and their conformalized forms (CQRF and CQRDNN, respectively)

Method	09:00	12:00	15:00	Mean
PINP 80% Prediction Interval				
QRF	**0.882**	**0.912**	**0.849**	**0.879**
	(0.310)	(0.298)	(0.245)	(0.284)
CQRF	**0.877**	**0.888**	**0.841**	**0.868**
	(0.307)	(0.294)	(0.241)	(0.281)
QRDNN	**0.830**	**0.874**	**0.849**	**0.821**
	(0.227)	(0.234)	(0.214)	(0.225)
CQRDNN	0.784	**0.827**	**0.816**	**0.809**
	(0.220)	(0.200)	(0.200)	(0.207)
PINP 85% Prediction Interval				
QRF	**0.921**	**0.937**	**0.885**	**0.914**
	(0.362)	(0.339)	(0.279)	(0.327)
CQRF	**0.912**	**0.929**	**0.877**	**0.906**
	(0.358)	(0.335)	(0.274)	(0.322)
QRDNN	**0.874**	**0.918**	**0.888**	**0.893**
	(0.261)	(0.266)	(0.242)	(0.256)
CQRDNN	0.822	**0.885**	**0.866**	**0.858**
	(0.244)	(0.231)	(0.228)	(0.234)
PINP 90% Prediction Interval				
QRF	**0.956**	**0.959**	**0.940**	**0.952**
	(0.438)	(0.397)	(0.325)	(0.387)
CQRF	**0.956**	**0.956**	**0.921**	**0.944**
	(0.434)	(0.393)	(0.320)	(0.382)
QRDNN	**0.904**	**0.945**	**0.904**	**0.918**
	(0.301)	(0.306)	(0.277)	(0.295)
CQRDNN	**0.882**	**0.934**	**0.904**	**0.907**
	(0.290)	(0.286)	(0.283)	(0.286)
PINP 95% Prediction Interval				
QRF	**0.978**	**0.984**	**0.970**	**0.977**
	(0.568)	(0.495)	(0.399)	(0.488)
CQRF	**0.975**	**0.978**	**0.967**	**0.974**
	(0.564)	(0.490)	(0.394)	(0.483)
QRDNN	0.948	**0.964**	0.934	0.949
	(0.361)	(0.362)	(0.324)	(0.349)
CQRDNN	0.942	**0.962**	0.948	**0.951**
	(0.369)	(0.342)	(0.352)	(0.354)
PINP 99% Prediction Interval				
QRF	**1**	**0.997**	0.989	**0.995**
	(0.821)	(0.688)	(0.546)	(0.685)
CQRF **1**	**0.997**	0.986	**0.995**	
	(0.816)	(0.684)	(0.541)	(0.680)
QRDNN	0.978	0.981	0.981	0.980
	(0.508)	(0.504)	(0.443)	(0.485)
CQRDNN	0.979	0.979	0.981	0.980
	(0.557)	(0.468)	(0.462)	(0.496)

In terms of the mean, the coverage is reduced when using the conformalized methods, but the resulting PICP values are not far from the target PINP, with the advantage that AIW is reduced in most cases. CQRDNN is generally the best performer in terms of AIW

Table 14 PICP and AIW (in parenthesis) results on the Córdoba data (solar energy) based on the time horizon for NGB (second best method on the Córdoba data) and QRDNN and their conformalized forms (CNGB and CQRDNN, respectively)

Method	09:00	12:00	15:00	Mean
PINP 80% Prediction Interval				
NGB	0.786	**0.849**	**0.836**	**0.824**
	(0.188)	(0.198)	(0.128)	(0.171)
CNGB	**0.819**	**0.830**	**0.860**	**0.837**
	(0.202)	(0.191)	(0.136)	(0.176)
QRDNN	**0.836**	**0.874**	**0.808**	**0.839**
	(0.252)	(0.188)	(0.143)	(0.194)
CQRDNN	**0.844**	**0.805**	**0.800**	**0.816**
	(0.179)	(0.171)	(0.140)	(0.164)
PINP 85% Prediction Interval				
NGB	0.838	**0.901**	**0.868**	**0.869**
	(0.211)	(0.222)	(0.144)	(0.193)
CNGB	**0.858**	**0.899**	**0.899**	**0.885**
	(0.221)	(0.219)	(0.164)	(0.201)
QRDNN	**0.866**	**0.934**	**0.880**	**0.893**
	(0.286)	(0.215)	(0.161)	(0.248)
CQRDNN	**0.877**	**0.874**	**0.863**	**0.871**
	(0.199)	(0.194)	(0.157)	(0.183)
PINP 90% Prediction Interval				
NGB	0.879	**0.942**	**0.904**	**0.909**
	(0.241)	(0.254)	(0.165)	(0.220)
CNGB	**0.921**	**0.942**	**0.926**	**0.930**
	(0.273)	(0.253)	(0.186)	(0.238)

Table 14 (continued)

Method	09:00	12:00	15:00	Mean
QRDNN	0.899	**0.964**	**0.915**	**0.926**
	(0.333)	(0.255)	(0.189)	(0.259)
CQRDNN	**0.921**	**0.948**	**0.910**	**0.926**
	(0.237)	(0.224)	(0.185)	(0.215)
PINP 95% Prediction Interval				
NGB	0.934	**0.975**	0.937	0.949
	(0.287)	(0.303)	(0.196)	(0.262)
CNGB	**0.959**	**0.984**	**0.964**	**0.969**
	(0.320)	(0.328)	(0.238)	(0.295)
QRDNN	0.943	**0.981**	**0.951**	**0.958**
	(0.400)	(0.306)	(0.223)	(0.310)
CQRDNN	**0.962**	**0.981**	**0.975**	**0.973**
	(0.295)	(0.324)	(0.238)	(0.286)
PINP 99% Prediction Interval				
NGB	0.970	**0.995**	0.978	0.981
	(0.378)	(0.398)	(0.258)	(0.344)
CNGB	**0.992**	**0.997**	**0.997**	**0.995**
	(0.497)	(0.459)	(0.413)	(0.456)
QRDNN	0.984	**0.995**	0.989	0.989
	(0.563)	(0.437)	(0.307)	(0.436)
CQRDNN	0.989	**0.992**	0.989	**0.990**
	(0.392)	(0.368)	(0.317)	(0.359)

In terms of the mean, CQRDNN is the best method, achieving PICPs closer to the target PINPs while reducing the AIW

We present the PICP and AIW results for every season, region, and PINP in Table 16.

In summary, the results based on the season follow the general trends displayed in the first part of this section, although specific behaviors are observed for some seasons. Generally, good coverage is achieved using CQRDNN with a few exceptions (e.g., the Granada data in the summer and winter, the Lugo data in the spring and summer (wind), and the Ciudad Real data in spring (solar)). However, the results remain close to the PINP. Relatively narrow intervals are still obtained using CQRDNN. For both solar regions, the narrowest intervals are obtained during the summer as this is the most stable season regarding solar radiation.

For the Granada data (wind energy), some PINP values in the summer and winter were low for CQRDNN. Regarding the width of the intervals, it can be seen that intervals for summer are wider, while in autumn, winter, and spring the AIW values are similar for equal values of the PINP. The analysis by season for the second wind energy region (Lugo) shows similar patterns, with the coverage being achieved, except for some PINP values in the spring and summer. However, the coverage remains close, especially for the summer results. Regarding AIW, the highest values are observed during the summer, while the narrowest intervals are obtained during winter and autumn.

For the solar energy regions, the results on the Ciudad Real data indicate a high coverage is achieved using CQRDNN during summer and for most PINP values during autumn. Coverage is lower during winter and spring. Regarding the seasons, the AIW during winter and autumn is high compared to spring and summer. Finally, for the Córdoba data, the required coverage is generally achieved by CQRDNN in each season. In this region, we cannot find significant differences for the AIW across the presented seasons.

In summary, the results based on season follow the general trends displayed in the first part of this section, although some specific behaviors are observed during some seasons. Generally, CQRDNN performs well with respect to coverage, with a few exceptions (Granada in summer and winter (wind), Lugo in spring and summer (wind), and Ciudad Real in spring (solar)). However, the results remain close to the PINP. Relatively narrow intervals are still achieved using CQRDNN. For both solar regions, the narrowest intervals are obtained during summer, as this is the most stable season regarding solar radiation.

Table 15 Coverage-width ratio improvement based on the use of conformalized quantile regression on DNNs (CQRDNN vs. QRDNN) for each region and for each target PINP

Region	80%	85%	90%	95%	99%
Granada (wind)	2.25%	3.24%	2.26%	1.76%	-0.55%
Lugo (wind)	4.57%	3.80%	5.80%	1.26%	-6.13%
Ciudad Real (solar)	3.65%	5.05%	1.68%	-1.33%	-1.89%
Córdoba (solar)	10.83%	12.35%	15.02%	5.70%	15.08%

Calibrated quantiles improve the ratio, except in a few cases for large PINP values

Table 16 Analysis of the PICP and AIW (in parentheses) values based on the season, region, and target PINPs for CQRDNN

Region	Season	PINP 80%	PINP 85%	PINP 90%	PINP 95%	PINP 99%
Granada (wind)	Winter	0.7880 (0.1188)	0.8300 (0.1359)	0.8653 (0.1557)	0.9326 (0.1895)	0.9849 (0.2778)
	Spring	**0.8241** (0.1093)	**0.8791** (0.1247)	**0.9176** (0.1426)	**0.9657** (0.1753)	**0.9918** (0.2575)
	Summer	0.7679 (0.1872)	0.8178 (0.2116)	0.8551 (0.2404)	0.9150 (0.2902)	0.9827 (0.4091)
	Autumn	**0.8101** (0.1283)	**0.8636** (0.1453)	**0.9072** (0.1657)	**0.9534** (0.2000)	**0.9902** (0.2844)
Lugo (wind)	Winter	**0.8280** (0.1023)	**0.8569** (0.1154)	**0.9051** (0.1360)	**0.9519** (0.1694)	**0.9931** (0.2766)
	Spring	0.7594 (0.1176)	0.8105 (0.1319)	0.8640 (0.1533)	0.9148 (0.1879)	0.9876 (0.2908)
	Summer	0.7919 (0.1591)	0.8490 (0.1782)	0.8910 (0.2059)	0.9402 (0.2509)	0.9880 (0.3773)
	Autumn	**0.8251** (0.1076)	**0.8702** (0.1213)	**0.9069** (0.1423)	**0.9577** (0.1763)	**0.9944** (0.2875)
Ciudad Real (solar)	Winter	**0.8022** (0.2828)	0.8425 (0.3154)	0.8974 (0.3785)	0.9507 (0.4589)	0.9744 (0.6236)
	Spring	0.7253 (0.2470)	0.7913 (0.2817)	0.8606 (0.3447)	0.9231 (0.4314)	0.9670 (0.6014)
	Summer	**0.9007** (0.1504)	**0.9468** (0.1787)	**0.9610** (0.2354)	**0.9858** (0.3146)	**1** (0.4933)
	Autumn	**0.8052** (0.2440)	0.8455 (0.2736)	**0.9064** (0.3290)	0.9438 (0.3996)	0.9738 (0.5455)
Córdoba (solar)	Winter	**0.8059** (0.2190)	**0.8755** (0.2470)	**0.9341** (0.2844)	**0.9717** (0.3745)	0.9890 (0.4771)
	Spring	**0.8351** (0.2595)	**0.8827** (0.2977)	**0.9157** (0.3448)	**0.9634** (0.4564)	**0.9927** (0.5518)
	Summer	0.7660 (0.2081)	**0.8972** (0.2452)	**0.9291** (0.2852)	**0.9574** (0.4008)	0.9894 (0.5341)
	Autumn	**0.8314** (0.2117)	**0.8727** (0.2330)	**0.9288** (0.2662)	**0.9700** (0.3316)	**0.9925** (0.4086)

Values in bold indicate that a PICP equal to or higher than the target PINP is achieved for this method. Regarding the PICP, differences across seasons are observed. For solar regions, the narrowest intervals are obtained in summer, which is the most stable season regarding solar radiation

6 Conclusions

In this work, deep neural networks (QRDNNs) were utilized to estimate multiple quantiles in the context of regional renewable energy production forecasting in Spain. These networks were compared with methods that have been used recently in the energy forecasting field, such as support vector quantile regression (SVQR), gradient boosting quantile regression (GBQR), natural gradient boosting (NGB) and random forests (RFs). The NWP variables were extracted from a spatial grid that encompasses the region and its extension for this purpose. Four regions were selected because they are representative of each type of renewable energy: Granada and Lugo for wind energy; and Ciudad Real and Córdoba for solar energy. Models were used to predict 10 conditional quantiles. The methodology involved systematic hyperparameter tuning by a grid search, where the best performing models were selected according to the mean quantile loss. In addition, from the 10 quantiles estimated, 5 PIs were constructed for different nominal probability coverage (80%, 85%, 90%, 95% and 99%). Both quantiles and intervals were evaluated by the appropriate metrics (quantile loss, CRPS, interval coverage and width (PICP and AIW, respectively), coverage-width ratio, and WS).

With respect to quantile estimation, the best performing method for the quantile loss metric for all regions, on average (across all time horizons), is QRDNN. This method is followed by QRF (wind) and by GBQR and SVQR (solar). Regarding CRPS, the lowest values are obtained using QRDNN and this time is followed by GBQR (wind) and NGB/SVQR (solar). In summary, QRDNN shows consistently good performance across both metrics and energy types/regions, whereas the performance of the other methods may depend on the metric and/or region.

Regarding PIs obtained from the quantiles and the coverage (PICP) and width (AIW) metrics, QRF and QRDNN are the two most consistent methods. The desired coverage (PINP) is always obtained (on average across all time horizons) using QRF in both solar and wind energy regions, whereas the PINP is obtained for most cases on average using QRDNN, and in any case, it never deviates from the desired value by more than 1%. An important advantage of QRDNN is that the intervals it generates are 8% to 29% narrower than the ones generated by QRF.

Another metric that displays the quality of QRDNN intervals is the WS. In solar energy regions, QRDNN is always the method with the lowest score, except for the PINP at 99% on the Ciudad Real data. For predicting wind energy, these results hold true for the majority of cases. Finally, concerning the ratio of PICP and AIW, QRDNN is always the best performing method, except on the Córdoba data (once the methods that are far away from the desired coverage are excluded).

In this work, conformalized quantile regression has been applied to further improve the quality of PIs. This is based on the calibration of the estimated conditional quantiles using a validation set. The general methodology has been extended by taking into account the time horizon of the prediction leading to an improvement in interval width. Overall, the conformalized version of QRDNN (CQRDNN) tends to perform better.

In summary, QRDNNs and especially their conformalized form, achieve consistently good performance across the different metrics, for both regional wind and solar electricity production, and are remarkable with respect to the narrowness of the generated PIs while offering good coverage.

Acknowledgements This publication is part of the I+D+i project PID2019-107455RB-C22, funded by MCIN /AEI / 10.13039 / 501100011033. This work was also supported by the Comunidad de Madrid Excellence Program.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data Availability Data employed in this article is available for free in ESIOS: Red Eléctrica España (https://www.esios.ree.es/), and ECMWF: ERA5 hourly data on single levels from 1979 to present (https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels)

Declarations

The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Torres-Barrán A, Alonso Á, Dorronsoro JR (2019) Regression tree ensembles for wind energy and solar radiation prediction. Neurocomputing 326:151–160
2. Van Der Meer DW, Widén J, Munkhammar J (2018) Review on probabilistic forecasting of photovoltaic power production and electricity consumption. Renew Sust Energ Rev 81:1484–1512
3. Van Der Meer DW, Shepero M, Svensson A, Widén J, Munkhammar J (2018) Probabilistic forecasting of electricity consumption, photovoltaic power generation and net demand of an individual building using gaussian processes. Appl Energy 213:195–207
4. Pinson P, Madsen H (2009) Ensemble-based probabilistic forecasting at horns rev. Wind Energy: Int J Progress Appl Wind Power Conversion Technol 12(2):137–155
5. Alessandrini S, Davò F, Sperati S, Benini M, Delle Monache L (2014) Comparison of the economic impact of different wind power forecast systems for producers. Adv Sci Res 11(1):49–53
6. Sadeghi S, Jahangir H, Vatandoust B, Golkar MA, Ahmadian A, Elkamel A (2021) Optimal bidding strategy of a virtual power plant in day-ahead energy and frequency regulation markets: a deep learning-based approach. Int J Electr Power Energy Syst 127:106646
7. Camal S, Michiorri A, Kariniotakis G (2019) Probabilistic forecasting and bidding strategy of ancillary services for aggregated renewable power plants. In: 6th international conference energy & meteorology
8. Benth FE, Di Persio L, Lavagnini S (2018) Stochastic modeling of wind derivatives in energy markets. Risks 6(2):56
9. Takeuchi I, Le Q, Sears T, Smola A (2006) Nonparametric quantile estimation. J Mach Learn Res 7:1231–1264
10. Meinshausen N, Ridgeway G (2006) Quantile regression forests. J Mach Learn Res, vol 7(6)
11. Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378
12. Duan T, Anand A, Ding DY, Thai KK, Basu S, Ng A, Schuler A (2020) Ngboost: natural gradient boosting for probabilistic prediction. In: International conference on machine learning. PMLR, pp 2690–2700
13. He Y, Li H, Wang S, Yao X (2021) Uncertainty analysis of wind power probability density forecasting based on cubic spline interpolation and support vector quantile regression. Neurocomputing 430:121–137
14. Dang S, Peng L, Zhao J, Li J, Kong Z (2022) A quantile regression random forest-based short-term load probabilistic forecasting method. Energies 15(2):663
15. Galván IM, Huertas-Tato J, Rodríguez-Benítez FJ, Arbizu-Barrena C, Pozo-Vázquez D, Aler R (2021) Evolutionary-based prediction interval estimation by blending solar radiation forecasting models using meteorological weather types. Appl Soft Comput:107531
16. Mitrentsis G, Lens H (2022) An interpretable probabilistic model for short-term solar power forecasting using natural gradient boosting. Appl Energy 309:118473
17. Liu H, Fang S, Zhang Z, Li D, Lin K, Wang J (2021) Mfdnet: collaborative poses perception and matrix fisher distribution for head pose estimation. IEEE Trans Multimedia:1–1. https://doi.org/10.1109/TMM.2021.3081873

18. Liu T, Wang J, Yang B, Wang X (2021) Ngdnet: nonuniform gaussian-label distribution learning for infrared head pose estimation and on-task behavior understanding in the classroom. Neurocomputing 436:210–220
19. Chai J, Zeng H, Li A, Ngai EW (2021) Deep learning in computer vision: a critical review of emerging techniques and application scenarios. Mach Learn Appl 6:100134
20. Pal SK, Pramanik A, Maiti J, Mitra P (2021) Deep learning in multi-object detection and tracking: state of the art. Appl Intell 51(9):6400–6429
21. Li Z, Liu H, Zhang Z, Liu T, Xiong NN (2021) Learning knowledge graph embedding with heterogeneous relation attention networks. IEEE Trans Neural Netw Learn Syst
22. Otter DW, Medina JR, Kalita JK (2020) A survey of the usages of deep learning for natural language processing. IEEE Trans Neural Netw Learn Syst 32(2):604–624
23. Guijo-Rubio D, Durán-Rosal A, Gutiérrez P, Gómez-Orellana A, Casanova-Mateo C, Sanz-Justo J, Salcedo-Sanz S, Hervás-Martínez C (2020) Evolutionary artificial neural networks for accurate solar radiation prediction. Energy 210:118374
24. Boubaker S, Benghanem M, Mellit A, Lefza A, Kahouli O, Kolsi L (2021) Deep neural networks for predicting solar radiation at hail region, saudi arabia. IEEE Access 9:36719–36729
25. Mellit A, Pavan AM, Lughi V (2021) Deep learning neural networks for short-term photovoltaic power forecasting. Renew Energy 172:276–288
26. Ogliari E, Guilizzoni M, Pretto S, Giglio A (2021) Wind power 24-h ahead forecast by an artificial neural network and an hybrid model: comparison of the predictive performance renewable energy
27. Khodayar M, Liu G, Wang J, Khodayar ME (2020) Deep learning in power systems research: a review. CSEE J Power Energy Syst
28. Mujeeb S, Alghamdi TA, Ullah S, Fatima A, Javaid N, Saba T (2019) Exploiting deep learning for wind power forecasting based on big data analytics. Appl Sci 9(20):4417
29. Torres J, Aguilar R, Zuniga-Meneses K (2018) Deep learning to predict the generation of a wind farm. J Renewable Sustainable Energy 10(1):013305
30. Ray B, Shah R, Islam MR, Islam S (2020) A new data driven long-term solar yield analysis model of photovoltaic power plants. IEEE Access 8:136223–136233
31. Bilgili M, Yildirim A, Ozbek A, Celebi K, Ekinci F (2021) Long short-term memory (lstm) neural network and adaptive neuro-fuzzy inference system (anfis) approach in modeling renewable electricity generation forecasting. Int J Green Energy 18(6):578–594
32. Mert İ (2021) Agnostic deep neural network approach to the estimation of hydrogen production for solar-powered systems. Int J Hydrog Energy 46(9):6272–6285
33. Ahmed Mohammed A, Aung Z (2016) Ensemble learning approach for probabilistic forecasting of solar power generation. Energies 9(12):1017
34. Voyant C, Motte F, Notton G, Fouilloy A, Nivet M-L, Duchaud J-L (2018) Prediction intervals for global solar irradiation forecasting using regression trees methods. Renewable Energy 126:332–340
35. David M, Luis MA, Lauret P (2018) Comparison of intraday probabilistic forecasting of solar irradiance using only endogenous data. Int J Forecast 34(3):529–547
36. Bakker K, Whan K, Knap W, Schmeits M (2019) Comparison of statistical post-processing methods for probabilistic nwp forecasts of solar radiation. Sol Energy 191:138–150
37. Cannon A (2018) Qrnn: quantile regression neural networks. R package version 2(3):0
38. Cannon AJ (2018) Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes. Stoch Environ Res Risk Assess 32(11):3207–3225
39. Cervone G, Clemente-Harding L, Alessandrini S, Delle Monache L (2017) Short-term photovoltaic power forecasting using artificial neural networks and an analog ensemble. Renew Energy 108:274–286
40. He Y, Li H (2018) Probability density forecasting of wind power using quantile regression neural network and kernel density estimation. Energy Conversion Manag 164:374–384
41. Romano Y, Patterson E, Candes E (2019) Conformalized quantile regression. Adv Neural Inf Process Syst:32
42. Hu J, Luo Q, Tang J, Heng J, Deng Y (2022) Conformalized temporal convolutional quantile regression networks for wind power interval forecasting. Energy 248:123497
43. Bessa RJ, Möhrlen C, Fundel V, Siefert M, Browell J, Haglund El Gaidi S, Hodge B-M, Cali U, Kariniotakis G (2017) Towards improved understanding of the applicability of uncertainty forecasts in the electric power industry. Energies 10(9):1402
44. Ozkan MB, Karagoz P (2021) Reducing the cost of wind resource assessment: using a regional wind power forecasting method for assessment. Int J Energy Res 45(9):13182–13197
45. Pierro M, Gentili D, Liolli FR, Cornaro C, Moser D, Betti A, Moschella M, Collino E, Ronzio D, Van Der Meer D (2022) Progress in regional pv power forecasting: a sensitivity analysis on the italian case study. Renew Energy 189:983–996
46. Khan M, Naeem MR, Al-Ammar EA, Ko W, Vettikalladi H, Ahmad I (2022) Power forecasting of regional wind farms via variational auto-encoder and deep hybrid transfer learning. Electronics 11(2):206
47. ECMWF: ERA5 hourly data on single levels from 1979 to present (2022) ECMWF: ERA5 hourly data on single levels from 1979 to present. https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels. Accessed 1 Jul 2021
48. ESIOS: red Eléctrica España (2022) ESIOS: red Eléctrica España. https://www.esios.ree.es. Accessed 1 Jul 2021
49. Koenker R, Portnoy S, Ng PT, Zeileis A, Grosjean P, Ripley BD (2018) Package quantreg. Reference manual available at R-CRAN: https://cran.rproject.org/web/packages/quantreg/quantreg.pdf. Accessed 1 Jul 2021
50. Drucker H, Burges CJ, Kaufman L, Smola A, Vapnik V (1996) Support vector regression machines. Adv Neural Inf Process Syst, vol 9
51. Steinwart I, Thomann P (2017) Liquidsvm: a fast and versatile svm package. arXiv:1702.06899
52. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y (2017) Lightgbm: a highly efficient gradient boosting decision tree. Adv Neural Inf Process Syst 30:3146–3154
53. Vasseur SP, Aznarte JL (2021) Comparing quantile regression methods for probabilistic forecasting of no2 pollution levels. Scientific Reports 11(1):1–8
54. Breiman L (2001) Random forests. Machine learning 45(1):5–32
55. Kumar M (2017) Scikit-garden: a garden for scikit-learn compatible trees. https://github.com/scikit-garden/scikit-garden. Accessed 1 Jul 2021
56. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) Pytorch: an imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d' Alché-Buc F, Fox E, Garnett R (eds) Advances in neural information processing systems 32. Curran Associates, Inc., ???, pp 8024–8035.

http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf. Accessed 1 Jul 2021
57. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization 3rd int. In: International conference on learning representations, Banff, Canada
58. Zamo M, Naveau P (2018) Estimation of the continuous ranked probability score with limited information and applications to ensemble weather forecasts. Math Geosci 50:209–234. Discussion started 21
59. Galván IM, Valls JM, Cervantes A, Aler R (2017) Multi-objective evolutionary optimization of prediction intervals for solar energy forecasting with neural networks. Inf Sci 418:363–382
60. Winkler RL (1972) A decision-theoretic approach to interval estimation. J Am Stat Assoc 67(337):187–191