Alcántara, A., Galván, Inés M., Aler, R. (2022). Direct estimation of prediction intervals for solar and wind regional energy forecasting with deep neural networks. *Engineering Applications of Artificial Intelligence*, 114, 105128.

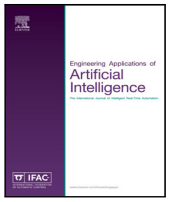
DOI: 10.1016/j.engappai.2022.105128

Engineering Applications of Artificial Intelligence 114 (2022) 105128

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Direct estimation of prediction intervals for solar and wind regional energy forecasting with deep neural networks

Antonio Alcántara, Inés M. Galván, Ricardo Aler *

Universidad Carlos III de Madrid, Avenida Universidad, 30, 28911, Leganés (Madrid), Spain

ARTICLE INFO

MSC:
68T05

Keywords:
Direct prediction intervals estimation
Quantile estimation
Deep neural networks
Hypernetworks
Regional renewable energy forecasting
Probabilistic forecasting

ABSTRACT

Deep neural networks (DNN) are becoming increasingly relevant for probabilistic forecasting because of their ability to estimate prediction intervals (PIs). Two different ways for estimating PIs with neural networks stand out: quantile estimation for posterior PI construction and direct PI estimation. The former first estimates quantiles, which are then used to construct PIs, while the latter directly obtains the lower and upper PI bounds by optimizing some loss functions, with the advantage that PI width is directly considered in the optimization process and thus may result in narrower intervals. In this work, two different DNN-based models are studied for direct PI estimation, and compared with DNN for quantile estimation in the context of solar and wind regional energy forecasting. The first approach is based on the recent quality-driven loss and is formulated to estimate multiple PIs with a single model. The second is a novel approach that employs hypernetworks (HN), where direct PI estimation is formulated as a multi-objective problem, returning a Pareto front of solutions that contains all possible coverage-width optimal trade-offs. This formulation allows HN to obtain optimal PIs for all possible coverages without increasing the number of network outputs or adjusting additional hyperparameters, as opposed to the first direct model. Results show that prediction intervals from direct estimation are narrower (up to 20%) than those of quantile estimation, for target coverages 70%–80% for all regions, and also 85%, 90%, and 95% depending on the region, while HN always achieves the required coverage for the higher target coverages.

1. Introduction

Probabilistic forecasting is currently attracting attention due to the fact that it is able to obtain high-value outputs that help to quantify the uncertainty of the predictions of a model. While classical point forecasting assigns a single value as a possible outcome, probabilistic forecasting associates a probability for possible events to occur. In classification tasks, probabilistic forecasting has been implicit in many cases. For example, logit models assign a probability to each of the two possible values of the dependent variable. This is not so simple in regression tasks, as we cannot give a probability to a single point in a continuous distribution. Constructing Prediction Intervals (PIs) is one of the main ways that the uncertainty can be quantified, by having lower and upper bounds where the dependent variable is contained with a certain probability.

Deep Neural Networks (DNN) have already shown their potential in probabilistic forecasting tasks. For example, in Hu et al. (2018), they were used for building PIs about the destination and time to reach a certain place for autonomous vehicles, outperforming other machine learning models like Quantile Regression Forests. In another case, the PIs generated by the networks were used to measure the uncertainty in landslide displacement (Lian et al., 2016).

Two main methodologies have gained weight in the state of the art of PI construction with DNN. Firstly, posterior PI estimation by means of quantile estimation is the most commonly used. Quantile estimation predicts the value for which the distribution function of the dependent variable is bigger than or equal to a certain value (the quantile). Then, it is statistically possible to derive PIs making use of these quantiles. The other methodology is to directly estimate the lower and upper bounds of PIs for the dependent variable. This is achieved by making use of a certain loss function that optimizes the two most relevant properties of PIs, width and coverage, with the aim of obtaining narrow intervals for each target coverage.

Quantiles can be estimated by minimizing quantile loss. Estimated quantiles can then be used to construct centered PIs. But this approach does not directly consider the width of PIs. With direct PI estimation, the coverage and the width of the PIs can be considered from the start, allowing the possibility of obtaining narrower intervals, especially because it allows PIs to be non-centered. The LUBE loss function (Khosravi et al., 2010) is an example of direct PI estimation. This loss is composed of two multiplicative terms: the width of the interval and a penalty term

* Corresponding author.
E-mail addresses: antalcan@est-econ.uc3m.es (A. Alcántara), igalvan@inf.uc3m.es (I.M. Galván), aler@inf.uc3m.es (R. Aler).

https://doi.org/10.1016/j.engappai.2022.105128
Received 18 October 2021; Received in revised form 20 May 2022; Accepted 22 June 2022
Available online xxxx

for cases when the required coverage is not achieved. However, LUBE has been typically optimized with evolutionary algorithms, which are usually very time consuming for complex problems. Recently, Pearce et al. (2018) derived the Quality-Driven loss function for PIs, which can be optimized by gradient descent and is therefore more efficient for neural network (and DNN) approaches.

A field where estimation of PIs with DNN is gaining acceptance is renewable energy production. However, in this field most of the studies found in the literature using neural networks estimate PIs from quantiles or directly PIs, while comparative studies of these two methodologies have not received much attention. Wan et al. (2013a,b) uses extreme learning machines (a simplified neural network) to directly construct PIs for wind forecasting, using an evolutionary approach, and compares them to some benchmarks, but not to quantile estimation. Similarly, in Khosravi and Nahavandi (2013), the LUBE approach with one hidden layer neural networks, is also used for direct PI construction for wind forecasting but compared only to (linear) quantile regression. The same approach is used in Li et al. (2019) for solar energy forecasting, where different evolutionary methods are compared, but only the direct estimation approach is tested. Also, in Galván et al. (2017), Aler et al. (2019) one-hidden-layer networks are optimized using multi-objective evolutionary methods, for directly building PIs for solar forecasting, but they were compared to quantile regression, Gradient Boosting Quantile regression or Quantile Random Forests, not to neural networks for quantile estimation. Li et al. (2020) uses deep networks for PI estimation, but only several direct PI estimation approaches are compared for wind power forecasting. This is also the case for the study in Liu et al. (2021), with a new LUBE approach for wind speed PI estimation. Other studies focus on quantile regression neural networks, such as Cannon (2018), where they are applied to predicting rainfall extremes. In He and Li (2018), Hatalis et al. (2017), David et al. (2018), Bakker et al. (2019) quantile neural networks are also used but the architectures applied contain only one or two hidden layers and the direct estimation of PIs is not addressed. As can be noticed, complete comparisons are not common when using neural networks: the majority of the studies focus only on quantile estimation or only on direct PI estimation. In addition, most of the works use neural network with one-hidden layer, but many-layers DNN optimized with gradient descent have been less explored.

In this article, we therefore compare the performance of quantile estimation (for posterior estimation of PIs) and direct PI estimation with DNN, with models estimating multiple quantiles/PIs. The experimental comparison will be carried out on renewable energy production aggregated at a regional level, for the two most important renewable sources (solar and wind). We are especially motivated by carrying out regional forecasting, as this is a less explored, but necessary application field (Bessa et al., 2017). For instance, works such as Wu et al. (2016) or Cervone et al. (2017) use DNN for posterior estimation of PIs, but only at a local, non-regional, level (several wind farms in China for the former and several solar plants in Italy for the latter).

Table 1 summarizes the literature review on neural network-based renewable energy PI estimation.“Probabilistic Forecasting Methodology” refers to whether the approach to probabilistic forecasting is quantile estimation for posterior PI estimation or direct PI estimation. This column represents one of the main topics of this article, the comparison of these two methodologies. The remaining columns qualify works in the literature regarding other issues at which our work aims to be more complete.“Single/Multiple Estimation” indicates whether single or multiple PIs/quantiles are estimated using a single model.“Use of DNN” refers to whether neural networks with more than two layers are allowed. The next column is “yes” when generated renewable energy production is forecast (rather than other variables such as solar radiation). It also qualifies whether solar, wind, or both are included in the study. “Regional context” evaluates whether energy is forecast for whole regions, and finally, “Results” provides a brief summary of the results presented in the literature. Our contribution is summarized at the bottom of Table 1. As it can be seen, quantile and direct PI estimation approaches are studied and compared, providing methods that estimate several PIs with the same model (including a novel approach). Our study uses DNN with more than two layers trained with gradient descent optimization. Also, the application domain is regional renewable energy production for both renewable sources: solar and wind.

In our work Quantile Regression Deep Neural Networks (QRDNN) have been used for the estimation of quantiles, by minimizing quantile loss for a set of quantiles. For direct PI estimation, two methods will be studied. The first one takes advantage of the improved loss recently introduced by Pearce et al. (2018) for neural networks. This method will be known as Quality-Driven loss Deep Neural Networks (QDDNN) in our work. The second approach for direct PI estimation is a novel method based on hypernetworks (HN) (Ha et al., 2016). HN are networks that generate the weights for another network, the so called main or target network, which is in charge of solving the task at hand (Ha et al., 2016). HN have been applied to multi-task learning problems where the aim is to learn several tasks simultaneously. HN are able to learn the whole Pareto front of solutions, where each point in the front represents a different performance trade-off between the multiple tasks (Navon et al., 2020). In our work, we use HN in a novel way, by formulating direct PI estimation as a multi-objective problem (maximization of PI coverage and minimization of PI width) and solving it with HN. The solution is a Pareto front which contains all possible coverage-width optimal trade-offs. That means that for every coverage (e.g. 70%, 80%, ...), the front contains a solution that returns intervals with that coverage and optimal width.

One important focus of our work for both quantile and direct estimation, is to obtain multiple quantiles or multiple PIs from a single model, as they are typically needed to characterize better the distribution of the response variable. For quantile estimation and for direct estimation with QDDNN, this is efficiently achieved by networks having multiple outputs (for the different quantiles or for the different PIs), and goes beyond other studies where only one PI is obtained (Pearce et al., 2018). In the case of PIs, this means that, for instance, there are two outputs (lower and upper bounds) for 70% coverage PI, another two for 80%, and so on. Our second method to direct PI estimation via HN, approaches the estimation of multiple PIs elegantly, because it allows PIs to be obtained for all possible coverages without multiplying the number of network outputs.

As mentioned, the empirical evaluation of the methods will be carried out in the probabilistic regional renewable forecasting field. To do so, DNN-based quantile models and direct PI estimation models (both QDDNN and HN) will be obtained for the electricity production in four provinces in Spain at different forecasting horizons. In order to have a broader understanding of deep networks for renewable probabilistic forecasting, this will be done for the two most important renewable energies: solar and wind. Provinces with high installed capacity for each of the energy sources will be used in the study: Ciudad Real and Córdoba provinces for solar, and Burgos and Lugo for wind. After an exhaustive grid-search hyperparameter selection, models will be evaluated by the quality of their PIs. The inputs to the model will be meteorological variable forecasts (Numerical Weather Prediction variables) defined on a grid that covers the region of interest, together with time-related variables, as solar and wind energy have direct dependence on these features.

To summarize, the novelty of this work with respect to the current state of the art is:

1. Comparing quantile estimation for PIs (i.e. posterior PI estimation) and direct PI estimation with DNN, an issue not widely addressed by the literature.
2. Studying a novel direct PI estimation method based on hypernetworks, which is able to generate a complete set of solutions for the PI width-coverage trade-off.

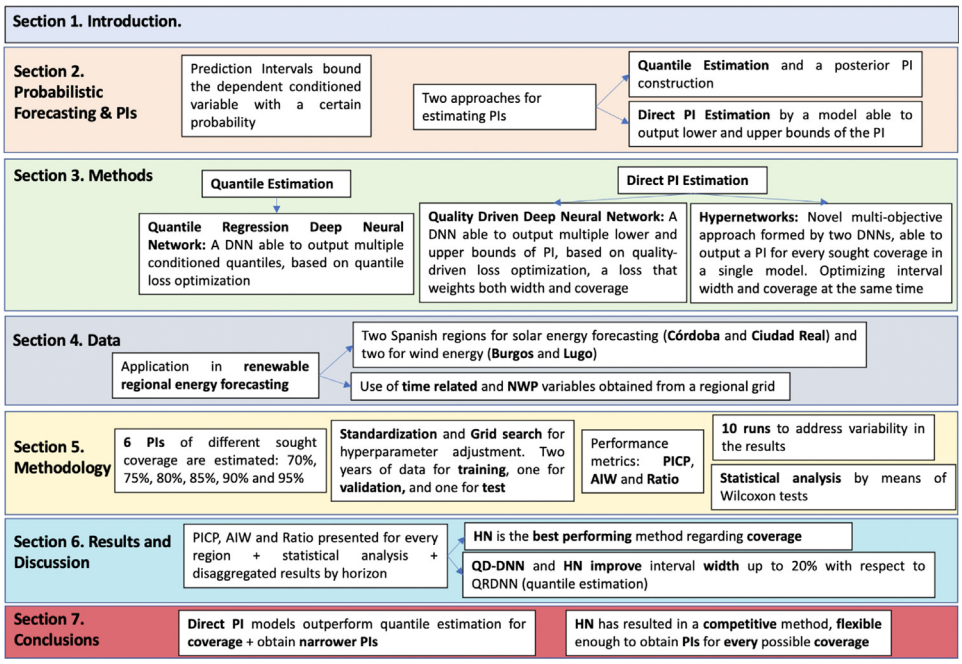


Fig. 1. Graphical view of the structure of our work.

3. Evaluating these methods in probabilistic forecasting of renewable energy production for both solar and wind energy, specifically in the regional forecasting context, a scope not yet widely considered.

The structure of this article is presented in the flow chart of Fig. 1. Section 2 describes in detail the two methodologies of probabilistic forecasting. Section 3 introduces the DNN based methods employed in this article. Section 4 describes the dataset used for empirical evaluation. Section 5 presents the methodology: models, metrics, and evaluation procedure. Section 6 presents and discusses the results obtained. Finally, Section 7 draws the main conclusions of our study.

2. Probabilistic forecasting and prediction intervals

One of the main approaches in probabilistic forecast tasks (that is, measuring the uncertainty of a prediction) is the construction of PIs. A PI for an output y depending on the inputs X is defined as the pair of lower and upper values p^{low} and p^{upp}, which contains the variable of interest with a certain probability. Formally, a PI is defined as a function G so that its outputs achieve a desired probability of including the target variable within the interval (Eq. (1)), also known as Prediction Interval Nominal Coverage (PINC), or target/required coverage.

$$G(X) = \left[p^{low}, p^{upp}\right], \text{ such that } P(p^{low} \leq y \leq p^{upp}) = PINC \tag{1}$$

Two metrics are usually used to evaluate a PI: the Average Interval Width (AIW) and the Prediction Interval Coverage Probability (PICP).

In Eq. (2) the PICP for a set of N_{ins} instances can be seen, where $\mathbf{1}$ is the indicator function of whether the target variable y is between the values of the PI generated by the function G (value 1), or not (value 0). This will basically measure the actual coverage in a set.

$$PICP = \frac{1}{N_{ins}} \sum_{i=1}^{N_{ins}} \mathbf{1}_{y_i \in G(\mathbf{X}_i)} \tag{2}$$

On the other hand, AIW is measured as the mean of the difference between the upper value p_i^{upp} and the lower value p_i^{low} of the PI and standardized by the possible values of the dependent variable, that is, the difference between the maximum value y_{max} and the minimum value y_{min} of the target variable (Eq. (3)).

$$AIW = \frac{1}{n(y_{max} - y_{min})} \sum_{i=1}^{n} \left(p_i^{upp} - p_i^{low}\right) \tag{3}$$

These two mentioned metrics are particularly involved in the two main objectives when creating a model for obtaining PIs:

- PICP ≥ PINC (actual coverage should achieve the required, target, or nominal coverage)
- AIW as small as possible

Normally, these two objectives participate in a trade-off: when the coverage increases, the width of the intervals also does, and when the objective of coverage is small, the PIs usually need a smaller width to accomplish the set goal.

In the state of the art, there are mostly two ways of obtaining PIs: a prior estimation of conditional quantiles followed by a posterior construction of PIs, or direct PI estimation.

The τ-quantile is defined such as the probability of Y being smaller than $Q_\tau(X)$ is τ (Eq. (4)).

$$Q_\tau(X) = \inf\{y : F(y|X = x) \geq \tau\} \tag{4}$$

where $F(y|X = x)$ is the cumulative distribution function of the conditioned dependent variable.

When the probabilistic forecast methodology is based on estimating the conditional quantiles, the model M can produce a set of N_{quant} quantiles (Eq. (5)).

$$M(X) = \hat{\mathbf{Q}}_\tau = \left[\hat{Q}_{(1)}, \hat{Q}_{(2)}, \ldots, \hat{Q}_{(N_{quan})}\right] \tag{5}$$

These outputs allow us to build PIs with a defined PINC from a statistical perspective. Let α be the probability of not covering the dependent variable ($\alpha = 1 - PINC$). PIs can be computed using conditional quantiles $\frac{\alpha}{2}$ and $1-\frac{\alpha}{2}$ as lower and upper bounds of the PIs, respectively. Notice how this approach will obtain centered PIs, leaving the same amount of probability to the left and to the right of the PI.

Therefore, the PI through conditional quantiles will be produced as shown in Eq. (6).

$$\hat{PI}_{1-\alpha}(X) = \left[\hat{p}_\alpha^{low}(X), \hat{p}_\alpha^{upp}(X)\right] = \left[\hat{Q}_{\frac{\alpha}{2}}(X), \hat{Q}_{1-\frac{\alpha}{2}}(X)\right] \tag{6}$$

Table 1
Literature review summary on neural network-based renewable energy PI estimation.

Reference	Probabilistic Forecasting Methodology	Single/multiple estimation	Use of DNN	Gradient descent approach	Dependent variable is forecast renewable energy production	Regional Context	Results
Aler et al. (2019)	Direct PI estimation	Multiple	No, one hidden layer for direct PI estimation	No, evolutionary approach	Yes, solar energy	No, several locations in Spain	Multi-objective approach outperforms baseline methods. The use of measured power as input improved PIs
Bakker et al. (2019)	Quantile estimation	Multiple	No, one hidden layer	Yes	No, solar radiation	No, selected stations	Quantile regression and random forests are generally the best performers
Cannon (2018)	Quantile estimation	Multiple	No, one hidden layer	Yes	No	Yes	The proposed MCQRNN model leads to more robust estimates of extreme rainfall
Cervone et al. (2017)	Quantile estimation	Single	Yes	Yes	Yes, only solar energy	No, three solar plants	A combined Analog Ensemble and Artificial NN yields the best results
David et al. (2018)	Quantile estimation	Multiple	No, one hidden layer	Yes	Yes, only solar energy	No, six different locations	None of the model combinations employed clearly outperformed the others
Galván et al. (2017)	Direct PI estimation	Multiple	No, one hidden layer	No, evolutionary approach	No, solar radiation	No, several locations in Oklahoma	The multi-objective approach provides similar or better results than the single-objective approach
Galván et al. (2021)	Direct PI estimation	Multiple	No, one hidden layer for direct PI estimation	No, evolutionary approach	No, solar radiation	No, single location	Quality of Prediction intervals is improved by using weather type information as input
Hatalis et al. (2017)	Quantile estimation	Multiple	No, one hidden layer	Yes	Yes, only wind energy	Yes, Ontario in Canada	The proposed NN approach leads to improved performance compared to baseline machine learning methods
He and Li (2018)	Quantile estimation	Single	No, one hidden layer	Yes	Yes, only wind power forecasting	No, two locations in Canada	PI coverage and width are improved through the proposed NN approach, in comparison to existing machine learning methods
Khosravi and Nahavandi (2013)	Direct PI estimation	Single	No, one hidden layer	No, evolutionary approach	Yes, only wind energy	No, several wind farms	The performance of the LUBE method for construction of NN-based PIs is enhanced by applying an ensemble method.
Li et al. (2019)	Direct PI estimation	Single	No	No, evolutionary	Yes, only solar energy	No, one location in Macau	The proposed NN model initialization approach performs better than the point prediction initialization and random initialization approaches
Li et al. (2020)	Direct PI estimation	Single	Yes	Yes	Yes, only wind energy	No, different locations in the US	The new deep LUBE model proposed obtains a large improvement over its competitors for estimating a single PI
Liu et al. (2021)	Direct PI estimation	Single	Yes	Yes	No, wind speed forecasting	No	Improving LUBE model with a Huber loss function reduces the training time and improves the quality of the PI
Wan et al. (2013a,b)	Direct PI estimation	Single	No, extreme learning machines	No, evolutionary approach	Yes, only wind energy	No, several wind farms	The proposed ELM approach outperformed benchmark methods regarding efficiency and reliability
Wu et al. (2016)	Point forecasting and posterior PI construction by error estimation	Multiple through error distribution	Yes	Yes	Yes, only wind energy	No, different wind farms	The DNN short-term point forecasting model and the posterior PI construction by error estimation achieve better results compared with traditional methods
This article	Quantile estimation and Direct PI estimation +novel HyperNetwork approach	Multiple	Yes, in all cases	Yes	Yes, both solar and wind energy	Yes, four Spanish regions	DNN-based models for direct estimation of PIs outperform quantile estimation. HN-based method obtains competitive results for estimating a complete set of PI solutions

On the other hand, through direct PI estimation methodology, the lower and upper bounds of the PI are directly estimated, allowing the possibility of constructing non-centered intervals. Besides, one or several p PIs can be obtained simultaneously as the outputs of a model M (Eq. (7)). Each of these intervals are defined by an α, the complementary of the nominal coverage as explained above.

$$M(X) = \mathbf{P\hat{I}_{\alpha}} = \left[\hat{PI}_{\alpha_1}, \hat{PI}_{\alpha_2}, \ldots, \hat{PI}_{\alpha_p}\right] = \left[\left(\hat{p}^{low}_{\alpha_1}, \hat{p}^{upp}_{\alpha_1}\right), \ldots, \left(\hat{p}^{low}_{\alpha_p}, \hat{p}^{upp}_{\alpha_p}\right)\right] \tag{7}$$

3. Methods

In the upcoming section, the conditional quantile estimation and direct PI estimation methodology with DNN will be discussed. Furthermore, a novel method based on HN will be presented in order to obtain the complete Pareto front of solutions for the trade-off of coverage and width in PIs.

These methods will be trained for the final goal of obtaining PIs. Aside from using continuous features as most fully connected deep networks do, embedding layers are included in the methods for the correct use of categorical variables.

An embedding is a learned continuous low-space representation of a given feature, which can be added to the inputs of a DNN as other continuous variables. Most of the work related with embeddings has been done in a natural lenguage processing context, but is also useful for categorical features, helping to obtain relationships between the categories and increasing the performance of deep learning models.

Take as an example a categorical feature with V possible values. The one-hot encoding representation of its values will be a vector of size $1 \times V$. This vector gets a transformation through a multiplication with the learned embedding weight matrix of size $V \times E$, where E is selected by the user to reduce the dimension. This will return an embedded vector of size $1 \times E$. Embedding layers (weights) are initialized with random values, and then updated through backpropagation, as every other weight in the network is updated (Guo and Berkhahn, 2016).

3.1. Quantile regression deep neural networks

DNN are well-known powerful and flexible methods. Their training phase is based on gradient descent and the use of different loss functions will allow a wide range of possible outcomes. As mentioned before, one of the main approaches in probabilistic forecasting is the estimation of conditional quantiles. To obtain these outputs with DNN, the quantile loss has to be employed, giving the method the name of Quantile Regression Deep Neural Networks (QRDNN).

This loss is shown in Eq. (8), where u represents residuals $y - \hat{Q}_{\tau}(\mathbf{X})$.

$$Loss_{\tau}(u) = \begin{cases} \tau u, & u \geq 0 \\ (\tau - 1)u, & u < 0 \end{cases} \tag{8}$$

Therefore, given a set of N_{ins} instances $T = \{(\mathbf{x}_i, y_i)_{i=1}^{N_{ins}}\}$, the τ-quantile loss is defined as the mean of losses over the entire set.

$$Loss_{\tau}(T) = \frac{1}{N_{ins}} \sum_{i=1}^{N_{ins}} Loss_{\tau}(y_i - \hat{Q}_{\tau}(\mathbf{x}_i)) \tag{9}$$

To avoid the loop of checking whether the residual of an instance is positive or negative, a more efficient implementation of the quantile loss (Eq. (10)) was built employing matrix operations. Thus,

$$Loss_{\tau}(T) = \frac{1}{N_{ins}} \sum \max(\tau U_{\tau}, (\tau - 1)U_{\tau}) \tag{10}$$

where U_{τ} is the column vector representing the residuals calculated over all the training set, that is $y - \hat{\mathbf{Q}}_{\tau}(\mathbf{x})$. The max operation returns another column vector $(\max(\tau u_{\tau,1}, (\tau - 1)u_{\tau,1}), \max(\tau u_{\tau,2}, (\tau - 1)u_{\tau,2}), \ldots)^T$. As τ is a non-negative value between zero and one, the max operator will return τu_{τ} when the residual is positive and $(\tau - 1)u_{\tau}$ when it is negative. After that, all the values in the column max vector will be added and weighted by the number of instances N_{ins} in the set, obtaining the τ-quantile loss. This kind of implementation will allow a faster training phase and the use of GPUs within the Pytorch (Paszke et al., 2019) framework: the one used in our work.

The structure of this DNN can be compared to the typical structure presented in most of the fully connected networks. First, an input layer with the input predictors. These predictors can be continuous or categorical (after having been tranformed through embedding layers). Next, several hidden layers will be implemented, as well as a final output layer to obtain different conditional quantiles.

Notice that each hidden layer is built with sequential layers: a fully connected layer, followed by a non-linear activation layer (ELU, ReLU, Sigmoid, Tanh, ...) and possibly a dropout layer to avoid overfitting. On the other hand, the output layer is important for this method, as it will produce multiple quantiles that will enable building several PIs with different coverages. This structure can be seen in Fig. 2.

As mentioned above, the quantile loss is calculated for just one quantile. Therefore, as the objective is to obtain several quantiles $\boldsymbol{\tau} = (\tau_{(1)}, \tau_{(2)}, \ldots, \tau_{(N_{quan})})$ with the same model, there will be an output for every τ-quantile, and a mean quantile loss across all the estimated quantiles (Eq. (11)).

$$Loss_{\boldsymbol{\tau}}(T) = \frac{1}{N_{quan}} \sum_{q=1}^{N_{quan}} Loss_{\tau_q}(T) \tag{11}$$

Therefore, from a set of N_{quan} quantiles $\boldsymbol{\tau}$, the quantile loss (Eq. (10)) will be computed for each one of them. Later it will be averaged across them (Eq. (11)) and employed in the backpropagation process.

As in the majority of studies done with deep networks, hyper-parameter tuning plays a crucial role in the correct performance of the models. In QRDNN, typical hyper-parameters must be tuned, like the type of activation layer (ELU, ReLU, Sigmoid, Tanh...), how many hidden layers to use and the number of neurons in each layer, the batch size, the learning rate or the optimizer (Stochastic Gradient Descent is mainly used, but Adam also gives a good performance (Kingma and Ba, 2014)).

When the training process ends, PIs can be built from their corresponding conditional quantiles, as explained in Section 2.

3.2. Direct prediction interval estimation with deep neural networks

The other methodology for obtaining PIs (and the main purpose of this article) is direct PI estimation using DNN. This means that, in our work, the outputs from the model will be the lower and upper bounds of several PIs.

In order to build models that directly generate PIs, the width and the coverage of the PIs must be taken into account in the training process, establishing a trade-off between both metrics. A loss with these characteristics was introduced in Pearce et al. (2018): the so-called Quality Driven Loss (QD-Loss) (Eq. (12)):

$$Loss_{QD-\alpha} = AIW_{capt.} + \lambda \frac{b}{\alpha(1-\alpha)} \max(0, (1-\alpha) - PICP))^2 \tag{12}$$

This loss is the summation of two terms related to the width and the coverage of the PIs (instead of a multiplication, in order to avoid a possible minimum where PIs are of zero width (Pearce et al., 2018)). The width of the interval is only considered in cases where the value of the dependent variable is captured, so that the size of intervals not covering the response variable, is not taken into account. The second term of Eq. (12) penalizes quadratically intervals that cover less than required ($PICP < PINC$). Notice that $\max(0, (1-\alpha) - PICP)) = \max(0, PINC - PICP))$, which is zero if the nominal coverage is satisfied ($PICP \geq PINC$), and the difference between the nominal and the actual coverage otherwise ($PINC - PICP$). λ sets the balance between the width and coverage goals. The b parameter represents the number of instances of the evaluated set (or the batchsize when training) to weight positively larger data sizes.

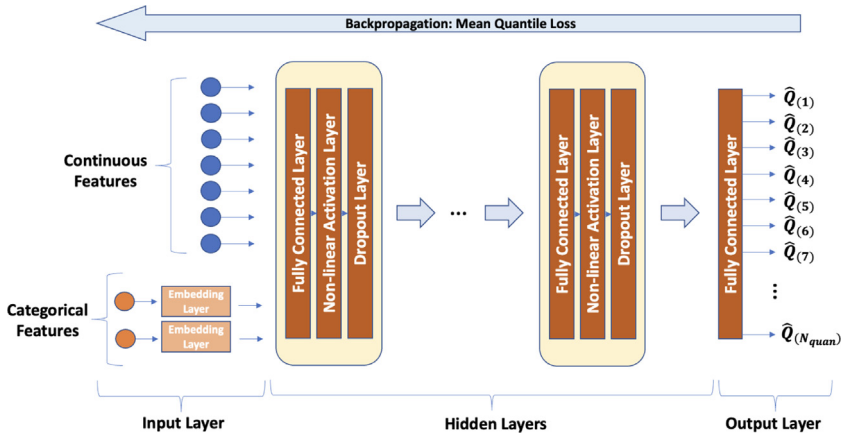


Fig. 2. Implemented Quantile Deep Neural Network structure.

Some problems arise with the current form of the loss explained above, and the minimum may not be reached by gradient descent due to discontinuities in the PICP. Thus, the soft version of the QD-loss has been used as in Pearce et al. (2018) and is shown in Algorithm 1, employing the sigmoid function and a softening factor for computing the PICP.

The QD-Loss will be implemented in DNN for obtaining high-quality PIs. The network will need at least two outputs, a lower and an upper bound, in order to compute the loss. However, one focus of our work is to obtain several PIs with the same model. Thus, our approach is based on the QD-Loss but considering multiple PIs, where p PIs will be estimated, resulting in $2p$ outputs from the neural network. Therefore, the QD-loss will be computed for every PI in the output and averaged over the p losses, as in Eq. (13).

$$Loss_{QD-\boldsymbol{\alpha}} = \frac{1}{p}\sum_{i=1}^{p} Loss_{QD-\alpha_i} \tag{13}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_p)$ is the complementary vector of the p PINC values.

The algorithm for computing the mean QD-loss across all the PIs (Eq. (13)) is shown in Algorithm 1. In practice, a softening factor s of 160 works correctly.

Algorithm 1: Implementation of the Mean QD-Loss.

Data: Target values $\mathbf{y}$, lower $\hat{\mathbf{p}}_{\boldsymbol{\alpha}}^{low}$ and upper $\hat{\mathbf{p}}_{\boldsymbol{\alpha}}^{upp}$ bounds for the different PIs, size p complementary vector $\boldsymbol{\alpha}$ of sought PINCs, softening factor s, parameter of coverage importance λ and number of observations or batch size b. ($\odot$ denotes the element-wise product)

Result: $Loss_{QD-\boldsymbol{\alpha}}$

for $i \in \{1 : p\}$ **do**

$K_{HU,i} = max(0, sign(\hat{\mathbf{p}}_{\alpha_i}^{upp} - \mathbf{y}))$

$K_{HL,i} = max(0, sign(\mathbf{y} - \hat{\mathbf{p}}_{\alpha_i}^{low}))$

$K_{H,i} = K_{HU,i} \odot K_{HL,i}$

$K_{SU,i} = sigmoid((\hat{\mathbf{p}}_{\alpha_i}^{upp} - \mathbf{y})s)$

$K_{SL,i} = sigmoid((\mathbf{y} - \hat{\mathbf{p}}_{\alpha_i}^{low})s)$

$K_{S,i} = K_{SU,i} \odot K_{SL,i}$

$AIW_{capt.,i} = \text{reduce-sum}((\hat{\mathbf{p}}_{\alpha_i}^{upp} - \hat{\mathbf{p}}_{\alpha_i}^{low}) \odot K_{H,i})/\text{reduce-sum}(K_{H,i})$

$PICP_{S,i} = \text{reduce-mean}(K_{S,i})$

$Loss_{QD-\alpha_i} = AIW_{capt.,i} + \lambda \frac{b}{\alpha_i(1-\alpha_i)} \max(0, (1-\alpha) - PICP_{S,i}))^2$

$Loss_{QD-\boldsymbol{\alpha}} = \frac{1}{p}\sum_{i=1}^{p} Loss_{QD-\alpha_i}$

It is important to notice how this loss function has its own parameters to tune: λ, and the batchsize b (quantile loss on the other hand, had no extra parameters). Besides, a specific value of λ or b must be fixed for a proper comparison in the evaluation process. In this sense, the loss resulting from a training process with λ_1 cannot be compared with the results of another training with λ_2. For this reason, the value of λ and n has to be evaluated during the hyper-parameter tuning process, as Section 5 will address.

The structure of the neural network to estimate PI is similar to the quantile network. The input layer can use continuous features or categorical ones passed through embedding layers. The hidden layers are also sequential operations of fully connected layers, non-linear activation layers and dropout ones. On the other hand, the outputs from the output layer will be pairs of low and upper bounds to conform p different PIs, as shown in Fig. 3.

In the training process, several hyper-parameters (similar to the quantile networks) should be tuned, such as the number of layers and neurons, the learning rate, the optimizer or the type of activation layer.

3.3. Learning the prediction interval pareto front with hypernetworks

The previous section showed how to obtain PIs using a single-objective loss function, where the two objectives (width and coverage penalty) are linearly combined, using λ as a tradeoff parameter.

In this section, we intend to formulate the direct estimation of PIs as a multi-objective problem, instead of aggregating the two objectives, as in the previous section. This can be formulated as minimizing AIW while also minimizing $\epsilon = 1 - PICP$. Notice that ϵ is the complementary of the PI actual coverage (as opposed to $\alpha = 1 - PINC$, which is the complementary of the required or nominal coverage). The goal is to obtain the set of points in loss-space that minimize one loss without making the other worse. These points will be called Pareto Optimal, and the set of all these points will be the Pareto front, that is, the solution of the problem.

Most of the methods that deliver a Pareto front for neural networks have been based on evolutionary computation techniques (Galván et al., 2017), due to the difficulty of using gradient descent in a multi-objective context. However, recently developed methods employing hypernetworks (HN) (Ha et al., 2016; Navon et al., 2020) have enabled this option, avoiding the large computational cost that would involve evolutionary techniques when a large number of weights must be optimized, as in the case of DNN. Hypernetworks have been employed in a variety of multi-objective problems, such as Multi-task regression, multi-MNIST (image classification), or pixel-wise classification and regression (Navon et al., 2020), but not for estimating PIs, as far as we know.

Let us first consider the PI multi-objective problem with the objective vector $\boldsymbol{l} = (AIW, \epsilon = 1 - PICP)$. Thus, both objectives are to be minimized so that, for every coverage $\epsilon = 1 - PICP$, the minimum AIW is obtained. This way, the final Pareto front of solutions

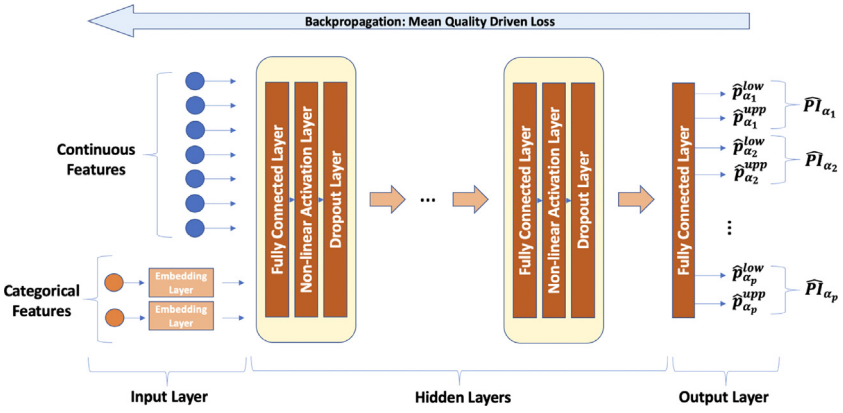


Fig. 3. Structure of Deep Neural Network for direct prediction interval estimation.

$(AIW, \epsilon = 1 - PICP)$ will contain the solutions with optimal trade-off between width and coverage. That means that it contains points (AIW, ϵ) such that for every $\epsilon = 1 - PICP$, there is a solution with optimal AIW (i.e. the narrowest interval found for that coverage). Thus, if a nominal coverage $PINC$ is required, looking in the Pareto front for the point $(AIW, 1-PINC)$, the solution with optimal PI width will be found. Fig. 5 visualizes a Pareto front where each red point corresponds to a neural network with the desired properties (AIW and PINC).

The HN approach is actually based on two neural networks: one small network (also known as the hypernetwork) that will generate the weights for the other network (called the main or the target network), which is in charge of generating the PIs.

The hypernetwork takes as input a vector of preferences between the two objectives $\boldsymbol{r} = (r_{AIW}, r_\epsilon)$, such as $\sum_j r_j = 1$. The preferences $\boldsymbol{r}$ correspond to a point in the Pareto front. The hypernetwork maps $\boldsymbol{r}$ into a high-dimensional space using a multilayer perceptron (MLP). The outputs of the hypernetwork are in fact the weights of the target network, which is a solution to the preference vector $\boldsymbol{r}$. For instance, large r_{AIW} should result in target networks that generate narrow PIs with the best possible coverage for intervals that size. In other words, the hypernetwork actually learns the complete Pareto front, and different points in this front can be obtained by feeding the hypernetwork different preference vectors.

More formally, let $h(\cdot;\phi)$ denote the hypernetwork with parameters (weights) ϕ and $t(\cdot;\theta)$ the target network whose parameters are θ. The hypernetwork takes $\boldsymbol{r}$ as input, passes it through fully connected layers, and generates the weight matrices for the target network. In short: $h(\boldsymbol{r};\phi) = \theta_{\boldsymbol{r}}$. On the other hand, the target net $t(\mathbf{x};\theta_{\boldsymbol{r}})$ will pass the features $\mathbf{x}$ (continuous or categorical) through its fully connected and non-linear activation layers to generate upper and lower bounds for a PI. Thus, $t(\mathbf{x};\theta_{\boldsymbol{r}}) = \hat{PI}_{\boldsymbol{r}}$. This structure can be observed in Fig. 4.

To produce Pareto optimal solutions, a linear scalarization process is used. That is, the preference vector $\boldsymbol{r}$ will be used as the weights in a loss function: the loss to be minimized during backpropagation is the linear combination $l = r_{AIW} AIW + r_\epsilon \epsilon$. It should be noticed that only the hypernetwork weights $\boldsymbol{\phi}$ are trained by gradient descent (backpropagation). The target network always uses the weights $\boldsymbol{\theta_r}$ produced by the hypernetwork. During the training process, the preference vector $\boldsymbol{r}$ is sampled randomly from a Dirichlet distribution of parameter $\beta \in R^2$ in every epoch. This allows the hypernetwork to find a general mapping (with weights $\boldsymbol{\phi}$) from $\boldsymbol{r}$ to $\boldsymbol{\theta_r}$ (which in turn produces a general mapping from $\boldsymbol{x}$ to $\hat{PI}_{\boldsymbol{r}}$, via the target network with weights $\boldsymbol{\theta_r}$).

While the hypernetwork allows target networks to be generated for any given preference vector, $t(\cdot;h(\boldsymbol{r};\phi))$, which in turn generates PIs with some PICP coverage and width, typically users need to go the other way around: first the user defines the desired nominal coverage (PINC), and the appropriate target network with PICP $\approx$ PINC should be obtained, by obtaining the corresponding $\boldsymbol{r}$ preference vector. However, the relation between the desired PINC to the proper $\boldsymbol{r}$ is not straightforward to compute. In order to obtain $\boldsymbol{r}_{PINC}$, we have followed an empirical approach, by first creating a Pareto front using a validation set, (different from the training set used to train the hypernetwork in order to avoid overfitting). This validation Pareto front is constructed by feeding the hypernetwork a set of preference vectors $\{(cos(0°), sin(0°)), \ldots, (\cos(\gamma), \sin(\gamma)), \ldots, (\cos(90°), \sin(90°))\}$ for angles γ uniformly spaced from 0° to 90°. Users can choose the number of points n in this set of preference vectors. This set can be normalized $(\{\ldots, (\frac{\cos(\gamma)}{\cos(\gamma)+\sin(\gamma)}, \frac{\sin(\gamma)}{\cos(\gamma)+\sin(\gamma)}), \ldots\})$ so that the summation of preferences is 1. Then, the hypernetwork can be applied to each preference vector in the set, and the corresponding target networks can be obtained. This target network is then evaluated on the validation set, and the two metrics AIW and PICP computed. At the end of the process, a set VF that relates each preference vector to the corresponding point in the Pareto front, will be available (Eq. (14)).

$$VF = \{(\boldsymbol{r}_1, AIW_1, PICP_1), \ldots, (\boldsymbol{r}_v, AIW_v, PICP_v)\}. \tag{14}$$

Now, let us remember that our original problem was to recover the preference vector that generates a target network with some desired PINC: $\boldsymbol{r}_{PINC}$. It can be seen that VF can be used for that purpose. Let us suppose a network that generates PIs with coverage PINC is required. For that, a preference vector is needed for the HN. It will be selected as the one in the VF that achieves $PICP \geq PINC$ while minimizing the difference with the objective $|PICP - PINC|$. If the coverage goal is not accomplished by any point in the validation Pareto front, the point that minimizes $|PICP - PINC|$ is directly selected. In other words, the point of the validation Pareto front closest to the needed PINC is selected, but preferably one with $PICP \geq PINC$.

Fig. 5 illustrates a practical example of the selection process from the validation Pareto front. Each point of the validation front represents an AIW and 1-PICP (for simplicity) for a given preference vector. If the goal is to obtain a target network that estimates PIs with 80% coverage, the point in the VF with $1-PICP \leq 0.2$ while minimizing $|PICP - 0.2|$ is selected. As can be seen in the VF, this point corresponds with a $\epsilon = 1 - PICP$ of 0.19983 and AIW of 0.08811. Furthermore, by consulting VF (Eq. (14)), we are able to know that the preference vector that generated that result is $\boldsymbol{r} = [0.2515, 0.7485]$.

Algorithm 2 displays the general method for obtaining PIs from a desired PINC from VF: the preference vector $\boldsymbol{r}_{PINC}$ that generated the appropriate result in the VF will be selected and then given to the HN generating the required weights $\theta_{\boldsymbol{r}_{PINC}}$ for the target network. Then, this main network will take its inputs $\mathbf{x}$ and the imposed weights to generate estimations of PIs satisfying the PINC. It can be seen that the algorithm splits VF into D^+ and D^-, which are the points with PICP that are larger and smaller than PINC, respectively. r_{PINC} is obtained as the closest PICP in D^+ (PICP larger than required). Only in the unlikely

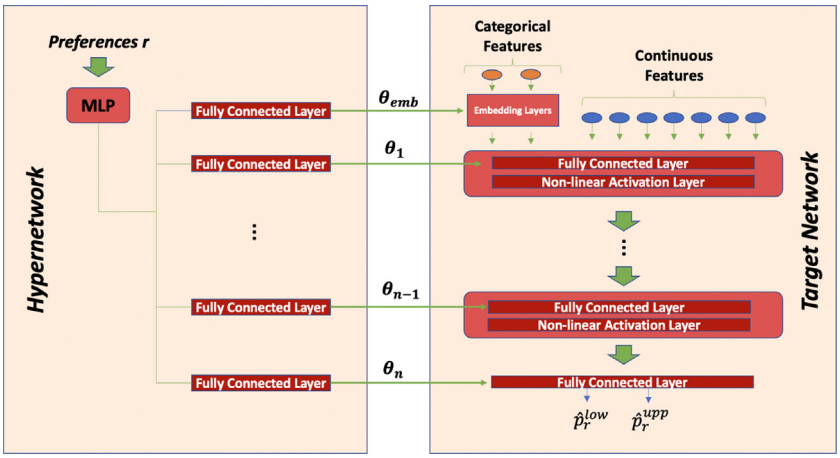


Fig. 4. Hypernetwork and target network structure for estimating PIs.

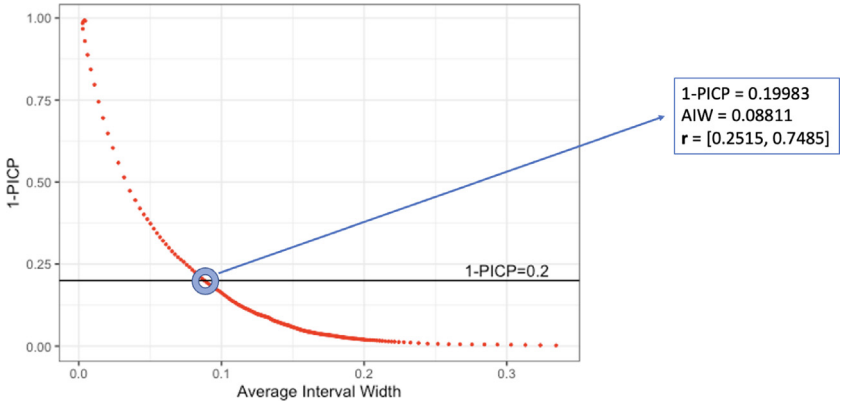


Fig. 5. Validation Pareto Front representation.

case in which no points in VF satisfy the required coverage, D^- is used. Another option for selecting the appropriate preference vector is to directly choose the one that minimizes $|PICP - PINC|$ (Galván et al., 2017). However, our modified implementation allows some solutions to be avoided where the PICP is closer to the PINC but not above it. In other words, we follow the heuristic that it is better to achieve the required PINC ($PICP \geq PINC$), even though the difference $|PICP - PINC|$ could be slightly larger than points which do not achieve the PINC ($PICP < PINC$), as one of the main objectives is to satisfy the desired coverage.

Algorithm 2: Preference vector selection and generation of prediction intervals from hypernetworks.

Data: $VF = \{(\boldsymbol{r}_1, AIW_1, PICP_1), ..., (\boldsymbol{r}_v, AIW_n, PICP_v)\}$, PINC
Result: $\hat{\boldsymbol{PI}}_{PINC}$

$D^+ \leftarrow \{(\boldsymbol{r}_k, AIW_k, PICP_k) \in VF | PICP_k \geq PINC\}$
$D^- \leftarrow \{(\boldsymbol{r}_k, AIW_k, PICP_k) \in VF | PICP_k < PINC\}$
if $D^+ \neq \emptyset$ **then**
  $\boldsymbol{r}_{PINC} \leftarrow \underset{(\boldsymbol{r}_k, AIW_k, PICP_k) \in D^+}{\text{argmin}} (|PICP_k - PINC|)$
else
  $\boldsymbol{r}_{PINC} \leftarrow \underset{(\boldsymbol{r}_k, AIW_k, PICP_k) \in D^-}{\text{argmin}} (|PICP_k - PINC|)$
end
$\theta_{\boldsymbol{r}_{PINC}} \leftarrow h(\boldsymbol{r}_{PINC}; \phi)$
$\hat{\boldsymbol{PI}}_{PINC} = t(\boldsymbol{x}; \theta_{\boldsymbol{r}_{PINC}})$

Finally, notice that, as for DNN in general, hyperparameters in the target network like the number of hidden layers, the number of neurons or the learning rate also have an important role for the appropriate performance of the method. The implementation of HN for PI estimation has been derived from the work done in Ruchte (2021).

4. Data

In order to predict the renewable energy generated (solar or wind) in a region, Numerical Weather Prediction (NWP) variables will be the inputs of our models. Regarding these independent variables, an observational spatial grid has been set across different Spanish regions ("provincias"), from which we will be able to obtain their values. This means the complete set of NWP variables will be collected at every point of the observational grid.

Data is provided by the European Centre for Medium-Range Weather Forecasts (ECMWF) within the ERA5 application in netCDF4 format. Overall, it is possible to obtain two data-products: ensemble mean (actual meteorological forecasts for each of the variables, provided as the mean of a forecast ensemble) and reanalysis data (posterior calibrations produced with the aim of reducing forecasting errors).

In our work, some preliminary experiments were carried out, suggesting that using the ensemble mean data allows a better modeling of the energy generation uncertainty. In this sense, models were trained with both ensemble mean data and reanalysis data, obtaining a better performance when ensemble data were used. Besides, this approach is closer to reality, because although reanalysis data can be used for training models, when using the model for making actual predictions, only meteorological forecasts (i.e. ensemble mean data) are available.

Ensemble mean data is given in a $0.5° \times 0.5°$ resolution every 3 h. Therefore, the dataset has been constructed with ensemble mean data, extracting the NWP variables from a spatial grid of $0.5° \times 0.5°$ resolution.

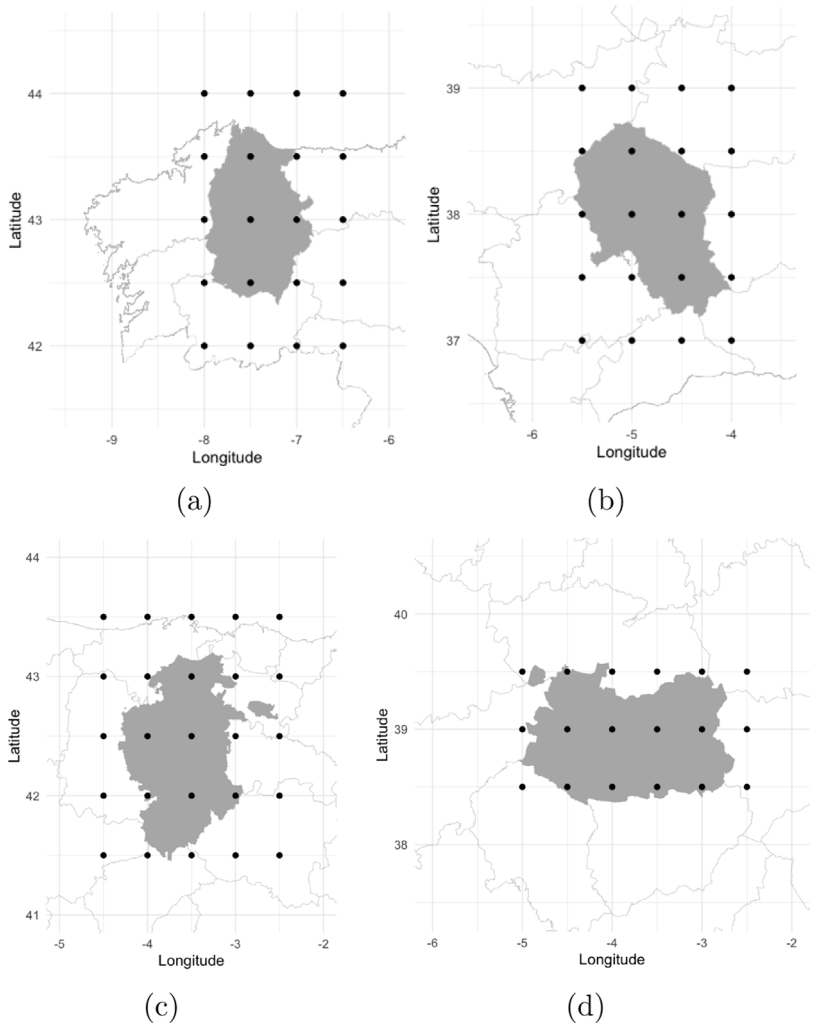


Fig. 6. (a) Observational grid for Lugo, (b) for Córdoba, (c) for Burgos, (d) for Ciudad Real.

Four grids are defined covering the majority of the extension of four regions (Spanish provinces). Grids on the regions of Córdoba and Ciudad Real will be employed for solar energy prediction. On the other hand, the ones in Lugo and Burgos will be used for wind energy (Fig. 6). The regions chosen for wind energy estimation are located in northern Spain, where wind conditions are prominent. At the same time, solar regions are located in the south, where the climate is warmer and radiation is higher. In all cases, a high generation capacity has been installed.

The grid for Lugo includes longitudes ranging from −8° to −6.5° and latitudes from 42° to 44°. For Córdoba, we get longitudes −5.5°/−4° and latitudes 37°/39°. For Burgos, the grid spans LON −4.5°/−2.5° and LAT 41.5°/43.5°. Finally, the grid on Ciudad Real comprises LON −5°/−2.5° and LAT 38.5°/39.5°.

On the other hand, the dependent variable (generated energy) is obtained from the Spanish regulator Red Eléctrica Española, within their open data portal, ESIOS (www.esios.ree.es). This tool allows the user to get data about energy consumption, generation, and exchange, among other indicators. Electricity generation data is given in hourly intervals. Besides, data can be filtered by the type of production (solar or wind in our design) and by region. Therefore, according to our region, we select the type of energy and the desired temporal set.

We will explain now how the complete dataset is built. A transformation is needed from the format data is provided by ECMWF to a 2-dimensional data matrix (with observations in the rows and variables in the columns). NWP variables, as provided by ECMWS, are contained in netCDF4 files, in a three dimensional array format: each variable will be measured at a specific latitude, longitude, and time. An arrangement is made in order to have every time point as an observation, and every different variable X_i in each latitude j and longitude k as our inputs. For example, if we have N meteorological variables in a $j \times k$ spatial grid, the procedure will allow us to get a set of T observations (rows) and $N \times j \times k$ independent variables (columns).

ECMWF provides forecasts for each variable, every day, for 8 forecast time horizons: 00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00, and 21:00 (hence, the temporal resolution of the ensemble mean data is 3 h). Therefore, there will be a maximum number of 8 observations per day in our dataset. The dependent variable (electrical energy produced) is obtained from the ESIOS system, by matching the time horizons of each observation with the times of the ESIOS system (e.g. ECMWF horizon 15:00 is matched with energy produced during the ESIOS period 15:00–16:00). For wind energy, all forecast horizons have been used. For solar energy, only those horizons which correspond to daylight for the whole year have been used (09:00, 12:00, and 15:00).

On the other hand, some time-related variables regarding the date of prediction are also included, as they could have a natural effect on the energy generation: forecast time horizon (00:00, 03:00, ..., 21:00), month (January, ..., December) and season (spring, summer, autumn and winter). These features will be treated as categorical, transformed with one-hot encoding and passed through embedding layers as explained in Section 5.

Table 2
Solar and wind energy prediction variables from the two sources: NWP (Numerical Weather Prediction) variables and time-related variables (forecast horizon, month, and season).

Features	Usage
NWP 100 m u-component of wind	Solar & Wind
NWP 100 m v-component of wind	Solar & Wind
NWP 100 m wind norm	Wind
NWP 10 m u-component of wind	Wind
NWP 10 m v-component of wind	Wind
NWP 10 m wind norm	Wind
NWP 2 m temperature	Solar & Wind
NWP Maximum 2 m temperature since previous post-processing	Solar
NWP Minimum 2 m temperature since previous post-processing	Solar
NWP Surface pressure	Solar & Wind
NWP Mean surface downward long-wave radiation flux	Solar
NWP Mean surface downward short-wave radiation flux	Solar
NWP Mean surface net long-wave radiation flux	Solar
NWP Mean surface net short-wave radiation flux	Solar
NWP Mean top downward short-wave radiation flux	Solar
NWP Mean top net long-wave radiation flux	Solar
NWP Mean top net short-wave radiation flux	Solar
NWP Total cloud cover	Solar
NWP Total precipitation	Solar
Forecast time horizon	Solar & Wind
Forecast month	Solar & Wind
Forecast season	Solar & Wind

Table 2 summarizes the features employed. The selection of NWP variables is made according to other research works in energy prediction using NWP variable grids (Martin et al., 2016; Andrade and Bessa, 2017; Torres-Barrán et al., 2019), while adapting to the availability in the ECMWF ERA5 open-access application. In general, variables related to radiation and temperature are employed for generated solar energy forecasting, whereas variables related to the state of the wind conditions were used for wind energy regions.

5. Methodology

5.1. Models

In this article, three different methods will be built with the final goal of obtaining multiple PIs of the renewable energy generated in a specific region: Quantile Regression Deep Neural Networks (QRDNN), Quality-Driven Loss Deep Neural Networks (QDDNN) and Hypernetworks (HN). Each method will take the NWP and time-related variables as inputs, and will be trained to obtain simultaneously six PIs with PINC values 70%, 75%, 80%, 85%, 90% and 95%, respectively.

For this purpose, QRDNN will produce 12 different quantiles in the output layer ($Q_{.025}$, $Q_{.05}$, $Q_{.075}$, $Q_{.1}$, $Q_{.125}$, $Q_{.15}$, $Q_{.85}$, $Q_{.875}$, $Q_{.9}$, $Q_{.925}$, $Q_{.95}$ and $Q_{.975}$) that will conform the respective PIs (see Eq. (6)).

5.2. Evaluation procedure

Three different datasets have being built for each of the four regions in order to train and evaluate the deep learning methods. Two full years of data have been used for training, one year for hyper-parameter tuning and evaluation, and another for testing. These periods of time have been chosen by knowing that the installed power capacity has not changed in the region, so we can train and test the models without further adaptations. As noted in Table 2, the time horizon information will be used jointly with the NWP variables.

Thus, the following datasets have been built for each of the regions:

- Lugo (Wind energy). Training set: years 2015 and 2016. Validation set: year 2017. Test set: year 2018. 163 inputs (20 grid points times 8 NWP variables plus 3 time-related variables).
- Burgos (Wind energy). Training set: years 2015 and 2016. Validation set: year 2017. Test set: year 2018. 203 inputs (25 grid points times 8 NWP variables plus 3 time-related variables).
- Córdoba (Solar energy). Training set: years 2016 and 2017. Validation set: year 2018. Test set: year 2019. 303 inputs (20 grid points times 15 NWP variables plus 3 time-related variables).
- Ciudad Real (Solar energy). Training set: years 2015 and 2016. Validation set: year 2017. Test set: year 2018. 273 inputs (18 grid points times 15 NWP variables plus 3 time-related variables).

Before the training process, some transformations must be done in our data. Firstly, NWP independent variables will be standardized, by computing the required standard deviation and mean from the training and validation sets (for each region), and using them on the training, validation, and test partitions. The same standardization was applied to the dependent variable (generated energy) after a logarithmic transformation.

Regarding the time-related variables, the three variables have been taken as discrete for applying an embedding transformation. Before that, a one-hot encoding processing has to be carried out. The season variable can take four possible values and will be embedded to a two values vector. The twelve possible months will allow us to get an embedded vector of six values. Finally, in wind energy regions, eight possible forecast horizons will create an embedding layer with output size of four. In solar regions; the three observations per day will make an embedded vector of size two. The output vectors of the embedding layer will join the NWP variables for the first fully connected layer in every method studied. Notice how the output dimension of an embedding layer has to be chosen. During our preliminary testing, half of the original dimension was chosen, allowing us to maintain at least two dimensions in the hour variable for solar regions and in the season variable for both regions, while at the same time obtaining good results.

In relation to the first model of direct PI estimation presented, QDDNN, some decisions about its loss have to be made beforehand. The main one is the value of λ, which weights the importance of achieving the desired coverage in the PIs. Setting this value in advance is important, as the value of two losses with different λ cannot be fairly compared. Besides, the best value for one PI might be different for another interval. For homogeneity reasons, the same value of λ was chosen between $j \times 10^{-i}$ with $j \in \{1, 2, 5\}$ and $i \in \{2, 3, 4, 5\}$ with some

Table 3
Hyper-parameters values taken into account for hyper-parameter selection.

Hyper-parameter space
Hidden layers: $3, 4, 5, 6, 7, 8$
Neurons per layer: $50, 100, 150, 200, 250$
Learning rate: $j \times 10^{-i}$ with $j \in \{1, 5\}$ and $i \in \{3, 4, 5, 6\}$
Optimizers: SGD & Adam

preliminary experiments for every PI in the output layer, with the idea of checking if the values are suitable. This preselection was made in order to achieve the required PINC in the validation set.

The same occurs with the batchsize, where high values have been set in each method and region to have more confidence in the PICP, which directly intervenes in the optimization of QDDNN and HN. For example, in QDDNN, as the batchsize is a parameter included in the loss, it makes losses with different batchsize configuration not comparable. Some values were preliminary tested, as 1000, 1500, 2000, 2500 and the number of instances in the training set. Selection was made as the minimum batchsize for which no computing problems appears in the training process, as a zero-divisor error might appear calculating the QD-loss when no point in the batchsize is captured by the PI. The error occurs at the moment of computing AIW for captured points (the mean of $K_{H,i}$ is zero, see Algorithm 1).

The use of ELU as non-linear activation layers has shown good results for our data. Lastly, 10% dropout layers were added in every hidden layer for QRDNN and QDDNN. However, it has not been used with HN, as keeping every weight generated in the HN for its use in the target network displayed a better behavior.

As explained before, the hyper-parameter selection is crucial for the correct performance of the different methods. A systematic grid-search has been carried out for the hyper-parameters displayed in Table 3. This has been achieved by training models with all possible combinations of hyper-parameter values on the training data, and using the validation data for evaluating and selecting the best hyper-parameter values combination. Regarding the number of training iterations (epochs), the validation loss has been computed for every epoch, and the model corresponding the optimal validation loss has been selected.

Regarding the loss, the average quantile loss across the 12 different quantiles (Eq. (11)) was employed for evaluating QRDNN models, while the mean Quality-Driven Loss of the six PIs (Eq. (13)) was used in QDDNN.

However, whereas QRDNN and QDDNN minimize a unique loss function for training and evaluation, this is not how HN works. As mentioned, during training, HN models will use a weighted loss of the two objectives, AIW and PICP, with preference vectors. During evaluation, the complete Pareto front has to be assessed and not just one point. For that reason, the validation front is built as explained in Section 3.3 and evaluated through the hyper-volume indicator (Zitzler and Thiele, 1998). This measures the quality of the non-dominated front by computing the volume (area, in our case) with respect to a certain reference point. As the Pareto front gets closer to the coordinates origin (that is, zero width and zero 1-PICP), the value of the hyper-volume increases. Therefore, for HN, the model with the highest hyper-volume of the validation Pareto front has been selected.

Table 4 shows the final models selected for each region, method and type of energy forecast.

5.3. Metrics

Three different metrics were employed in order to evaluate the quality of the PIs generated from each method in the test set, summarized in Table 5. Some of them have been already mentioned in Section 2.

Firstly, one of the most important metrics is PICP, which measures the proportion of instances covered by the PI and is given in Eq. (15), shown in Table 5. N_{ins} is the number of instances in the set, and $\mathbf{1}_{y_i \in \hat{PI}(\mathbf{x}_i)}$ is an indicator function that returns 1 when $y_i \in \hat{PI}(\mathbf{x}_i)$, and 0 otherwise. This is one of the most important metrics in probabilistic prediction, as one of the main goals is to achieve a PICP larger than or equal to the required coverage, PINC.

On the other hand, the Average Interval Width (Eq. (16) in Table 5) measures the width of the generated intervals, and is normalized for the maximum possible width in the set. $\hat{p}_i^{upp}(\mathbf{x}_i)$ and $\hat{p}_i^{low}(\mathbf{x}_i)$ represent the upper and lower bound of the $\hat{PI}(\mathbf{x}_i)$. If required coverage is accomplished, narrower PIs (small AIW values) will be preferred.

Given that coverage can be trivially increased by increasing interval width, the Ratio metric (Galván et al., 2021) between PICP and AIW (Eq. (17) in Table 5) measures the trade-off between the two metrics.

5.4. Stability analysis

To address the randomness in the results when training deep learning models with various initialized parameters, we implemented each model with ten different random seeds and calculated the standard error expressed via $\pm SE$ in Section 6 to reflect the model stability. Thus, low SE values will indicate how stable the model is when facing randomization in the training process. Results will always be presented as the mean from the ten different runs.

5.5. Validity analysis

Two different statistical tests will be employed to determine the quality of the PIs regarding the performance metrics:

- Statistical test for PICP: the non-parametric Wilcoxon signed rank single test (Wilcoxon, 1992) will be used to check whether the coverage is under the sought PINC.
- Statistical test for AIW and Ratio: statistical comparisons between methods by means of the two-samples Wilcoxon signed rank test will determine which PIs are narrower and which Ratio is bigger.

6. Results and discussion

In this section, the results of the different methods (QRDNN, QDDNN and HN) in each region are presented.

As explained before, the predictions of the renewable energy production are made for different time horizons. First, we will show the aggregated results of the methods as the average value of the metrics across the horizons. Next, AIW and Ratio will be broken down to check their evolution along the different hours.

6.1. Mean PI estimation performance

In Tables 6, 7, 8 and 9, the mean values of PICP, AIW and Ratio across time horizons are presented for Lugo, Burgos, Córdoba and Ciudad Real, respectively, and accompanied by their standard errors over ten different runs. Metrics are shown for every PI in terms of the sought PINC and for every model. The structure of these tables is defined as follows.

Regarding the PICP, an up arrow ↑ is displayed when the alternative hypothesis of PICP being smaller than the target PINC cannot be accepted under a 5% significance level (see Section 5.5). Besides, for direct PI estimation methods (QDDNN and HN), the average improvement in AIW and Ratio performance with respect to quantile estimation is added between brackets. In relation to the width and the ratio, further statistical comparisons between methods are shown in Table 10, and they will be explained later. However, it is important to comment at this point that the best result (of AIW or Ratio) in Tables 6 to 9 will be displayed in boldface when the corresponding method is statistically better than one of the other two methods (see Table 10). Otherwise no boldface will be used, meaning that the three methods perform similarly.

Table 4
Hyper-parameters selected for each region and method.

Method	Hyper-parameter	Lugo (wind)	Burgos (wind)	Córdoba (solar)	C. Real (solar)
QRDNN	Hidden layers	4	6	4	4
	Neurons per layer	50	150	250	200
	Learning rate	5×10^{-6}	5×10^{-6}	5×10^{-6}	5×10^{-6}
	Batch size	2500	2500	2000	2000
	Optimizer	Adam	Adam	Adam	Adam
QDDNN	Hidden layers	6	5	6	6
	Neurons per layer	100	150	100	150
	Learning rate	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}
	Batch size	2500	2500	2000	2000
	Optimizer	Adam	Adam	Adam	Adam
	Lambda	0.01	0.2	0.005	0.01
HN	Hidden layers	5	5	7	6
	Neurons per layer	150	200	150	200
	Learning rate	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}
	Batch size	2500	2500	2000	2000
	Optimizer	Adam	Adam	Adam	Adam

Table 5
Performance metrics used in the study.

Metric	Formula		Meaning
PICP	$\frac{1}{N_{ins}} \sum_{i=1}^{N_{ins}} \mathbf{1}_{y_i \in \hat{PI}(\mathbf{x}_i)}$	(15)	PI Coverage Probability
AIW	$\frac{1}{N_{ins}(y_{max} - y_{min})} \sum_{i=1}^{N_{ins}} \left[\hat{p}_i^{upp}(\mathbf{x}_i) - \hat{p}_i^{low}(\mathbf{x}_i)\right]$	(16)	Average PI Width
Ratio	$\frac{PICP}{AIW}$	(17)	Coverage-width trade-off

To begin with, aggregated results in Lugo (wind energy) are shown in Table 6. In terms of coverage, HN is the only method that achieves the goal for every PINC (QRDNN and QDDNN fail in 95%). Regarding the width of the intervals, direct estimation models are able to narrow the PIs between 10.4% and 3% with respect to quantile estimation for PINCs 70% to 85%. For the last two larger PINC values (90% and 95%), improvements are less clear for the AIW metric: for PINC 90% results are basically the same (as the statistical tests in Table 10 will confirm later), and for 95%, QRDNN has the lowest value but without achieving the coverage objective. Also, the Ratio improves with direct PI estimation for every PINC except 95%. In these cases, HN is always the best performing model. In relation to the variability of the results, SEs of the direct estimation methods (QDDNN and HN) are about half of the ones from quantile estimation (QRDNN). However, all SEs from the three models can be considered low.

Table 7 presents the results in the second wind energy region, Burgos. In this case, the SE behavior is similar to the one in Lugo, being low in all the methods, but especially for the direct PI estimation ones. Here, QRDNN is not able to satisfy the PINC goal from 85% to 95%, while direct estimation models produce intervals that satisfy this goal for every one of the six studied cases. Broadly speaking, HN obtain the best values for AIW and Ratio in PIs from 70% to 80% coverage, improving up to 12% in the smallest PINC value with respect to QRDNN. QDDNN also does, but to a lesser extent. For the largest PINC values, AIW and Ratio values are statistically the same for PINC 85% (see Table 10), whereas QRDNN performs better in AIW for 90% and in AIW and Ratio for 95%. However, this is at the cost of not satisfying the PINC goal, while direct estimation does.

The results of the first solar region, Córdoba, can be seen in Table 8, where models remain stable with low SE values. The differences in the SE of the results depend on the PINC, but SEs of direct estimation methods (QDDNN and HN) are always less than or equal to the SE for QRDNN. In this case, all the methods achieve the desired coverage for every PINC. Regarding the AIW, QDDNN is always the best performing method, followed by HN in every case (only for PINC 70%, AIW for QDDNN and HN are statistically the same). The percentage of improvement with respect to quantile estimation varies from 15% in PINC 70% to 6% in PINC 95%. In terms of the Ratio, HN obtain the same value for PINC 70% as QDDNN, while remaining second behind QDDNN for the rest of the intervals. In general, QDDNN always also outperform quantile estimation for the Ratio. The improvement ranges between 15% and 5% as well.

We will finish with the last solar region, Ciudad Real, in Table 9. Regarding the coverage, QRDNN fails in 95%, QDDNN also fails from 90% to 95%, while HN achieve the coverage for all the PINCs. With respect to the AIW, both QDDNN and HN take turns being the best performing method. HN is better for PINC values 70% and 75% and QDDNN for the rest of the intervals. The same behavior occurs with the Ratio. Roughly, direct PI estimation models are able to increase the quality of the intervals in terms of AIW and Ratio from almost 3% (with respect to quantile estimation) for the largest PINC value and more than 20% in the smallest PINC value. In relation to the SE, the differences in the results depend on the sought PINC, as in Córdoba, with HN being the method with the highest value, although with relatively low values for all models.

To finish with the analysis about the metrics on average, the models have been statistically compared regarding AIW and Ratio, for every region and PINC. Table 10 shows results of the Wilcoxon Signed Rank test. A + sign is shown when the first method makes an improvement in the metric over the second, a − sign in the opposite case, and a = sign if there is no statistical difference between both models. In all cases, a significance level of 5% was employed.

We will now discuss this statistical analysis by PINC. For PINC values 70%, 75% and 80%, direct estimation methods (QDDNN and HN) always outperform QRDNN for both AIW and Ratio in every region. Comparing HN with QDDNN for these PINCs, HN perform better

Table 6
Mean PICP, AIW and Ratio results in Lugo (wind energy).

PINC	Methodology	Model	PICP	AIW	Ratio
70%	Quantile estimation	QRDNN	0.731 ±0.014 ↑	0.114 ± 0.004	6.463 ± 0.116
	Direct PI estimation	QDDNN	0.725 ±0.005 ↑	0.104 ± 0.001 (9.1%)	7.112 ± 0.056 (10.0%)
		HN	0.742 ±0.006 ↑	**0.102 ± 0.002 (10.4%)**	**7.451 ± 0.133 (15.3%)**
75%	Quantile estimation	QRDNN	0.771 ±0.013 ↑	0.127 ± 0.005	6.152 ± 0.134
	Direct PI estimation	QDDNN	0.767 ±0.005 ↑	0.118 ± 0.002 (6.7%)	6.621 ± 0.057 (7.6%)
		HN	0.784 ±0.008 ↑	**0.117 ± 0.002 (7.7%)**	**6.888 ± 0.100 (12.0%)**
80%	Quantile estimation	QRDNN	0.814 ±0.013 ↑	0.141 ± 0.005	5.837 ± 0.129
	Direct PI estimation	QDDNN	0.812 ±0.006 ↑	0.134 ± 0.001 (4.8%)	6.160 ± 0.054 (5.5%)
		HN	0.826 ±0.007 ↑	**0.134 ± 0.002 (5.1%)**	**6.320 ± 0.055 (8.3%)**
85%	Quantile estimation	QRDNN	0.855 ±0.014 ↑	0.158 ± 0.006	5.461 ± 0.142
	Direct PI estimation	QDDNN	0.855 ±0.003 ↑	0.155 ± 0.001 (1.8%)	5.605 ± 0.043 (2.6%)
		HN	0.864 ±0.005 ↑	**0.154 ± 0.002 (2.9%)**	**5.738 ± 0.049 (5.1%)**
90%	Quantile estimation	QRDNN	0.896 ±0.012 ↑	0.181 ± 0.008	4.996 ± 0.151
	Direct PI estimation	QDDNN	0.898 ±0.003 ↑	0.182 ± 0.003 (−0.4%)	5.012 ± 0.069 (0.3%)
		HN	0.905 ±0.008 ↑	0.181 ± 0.005 (0.1%)	**5.084 ± 0.095 (1.8%)**
95%	Quantile estimation	QRDNN	0.943 ± 0.008	**0.219 ± 0.012**	**4.360 ± 0.195**
	Direct PI estimation	QDDNN	0.946 ± 0.002	0.227 ± 0.003 (−3.5%)	4.242 ± 0.046 (−2.7%)
		HN	0.952 ±0.002 ↑	0.228 ± 0.004 (−4.1%)	4.245 ± 0.068 (−2.6%)

Table 7
Mean PICP, AIW and Ratio results in Burgos (wind energy).

PINC	Methodology	Model	PICP	AIW	Ratio
70%	Quantile estimation	QRDNN	0.712 ±0.015 ↑	0.117 ± 0.003	6.174 ± 0.078
	Direct PI estimation	QDDNN	0.720 ±0.006 ↑	0.111 ± 0.002 (5.2%)	6.563 ± 0.083 (6.3%)
		HN	0.712 ±0.008 ↑	**0.104 ± 0.003 (10.8%)**	**6.911 ± 0.175 (11.9%)**
75%	Quantile estimation	QRDNN	0.760 ±0.011 ↑	0.130 ± 0.003	5.897 ± 0.081
	Direct PI estimation	QDDNN	0.764 ±0.006 ↑	0.125 ± 0.003 (4.2%)	6.170 ± 0.109 (4.6%)
		HN	0.760 ±0.007 ↑	**0.120 ± 0.004 (8.1%)**	**6.407 ± 0.176 (8.6%)**
80%	Quantile estimation	QRDNN	0.805 ±0.010 ↑	0.146 ± 0.005	5.567 ± 0.112
	Direct PI estimation	QDDNN	0.811 ±0.007 ↑	0.143 ± 0.004 (2.4%)	5.732 ± 0.109 (3.0%)
		HN	0.808 ±0.005 ↑	**0.139 ± 0.003 (5.2%)**	**5.877 ± 0.118 (5.6%)**
85%	Quantile estimation	QRDNN	0.844 ± 0.009	0.164 ± 0.005	5.211 ± 0.100
	Direct PI estimation	QDDNN	0.852 ±0.008 ↑	0.163 ± 0.004 (0.6%)	5.273 ± 0.089 (1.2%)
		HN	0.857 ±0.005 ↑	0.162 ± 0.003 (0.9%)	**5.323 ± 0.097 (2.1%)**
90%	Quantile estimation	QRDNN	0.889 ± 0.010	**0.189 ± 0.007**	4.758 ± 0.118
	Direct PI estimation	QDDNN	0.899 ±0.004 ↑	0.192 ± 0.003 (−1.7%)	4.723 ± 0.062 (−0.7%)
		HN	0.906 ±0.006 ↑	0.194 ± 0.004 (−2.8%)	4.713 ± 0.080 (−1.0%)
95%	Quantile estimation	QRDNN	0.939 ± 0.008	**0.231 ± 0.009**	**4.115 ± 0.120**
	Direct PI estimation	QDDNN	0.949 ±0.003 ↑	0.240 ± 0.004 (−4.1%)	3.990 ± 0.056 (−3.0%)
		HN	0.955 ±0.003 ↑	0.249 ± 0.006 (−8.0%)	3.872 ± 0.088 (−5.9%)

in wind energy regions (Lugo and Burgos). In Córdoba, QDDNN are better, whereas in Ciudad Real, HN improve over QDDNN for PINC 70% and 75%, and the opposite for PINC 80%. All three methods achieve the target coverage in the range 70% to 80% in the four regions (see Tables 6 to 9).

For PINC 85%, direct estimation methods improve AIW and Ratio over quantile estimation in Lugo, Córdoba and Ciudad Real, while obtaining similar results in Burgos. However, QRDNN does not reach the 85% target coverage in Burgos, while the two direct estimation methods achieve it (Table 7). No significant differences are observed between HN and QDDNN in Lugo and Burgos, whereas for solar regions, QDDNN perform better.

For PINC 90%, both direct estimation methods improve AIW and Ratio in Córdoba, with QDDNN being better than HN. In Ciudad Real, QDDNN outperform both QRDNN and HN, but at the cost of not achieving the coverage, as we saw before (Table 9). It is QRDNN and HN that achieve the target coverage in this case, and QRDNN obtain narrower intervals than HN. In Lugo, results among all the methods are relatively similar. In Burgos, QRDNN get narrower intervals, but at the cost of not achieving the 90% target coverage (the direct methods achieve it, see Table 7).

Finally, for PINC 95%, in Córdoba, QRDNN were outperformed by direct PI estimation methods, and QDDNN did better than HN for AIW and Ratio. In the other three regions, QRDNN sometimes obtain narrower intervals, but fail to achieve the target coverage. HN achieve it in the three regions while QRDNN only in Burgos (Table 7).

Summarizing results further, it is clear that both direct estimation methods perform better than quantile estimation for PINC values from 70% to 80% in all regions, in terms of PI width and Ratio. This is also true for higher PINC values, depending on the region (85% in Lugo and Ciudad Real, and 85%–95% in Córdoba). In two cases, results are similar (Lugo 90% and Burgos 85%). For high PINC values (90% and 95%) one of the direct methods, HN, has an advantage over QRDNN because HN always achieve the target coverage, while QRDNN does not in several cases (Lugo and Ciudad Real 95%; Burgos 90% and 95%, see Tables 6–9). This is the reason why QRDNN obtains narrower intervals

Table 8
Mean PICP, AIW and Ratio results in Córdoba (solar energy).

PINC	Methodology	Model	PICP	AIW	Ratio
70%	Quantile estimation	QRDNN	0.792 ±0.024 ↑	0.134 ± 0.005	5.950 ± 0.138
	Direct PI estimation	QDDNN	0.766 ±0.013 ↑	**0.114 ± 0.003 (14.9%)**	6.850 ± 0.145 (15.1%)
		HN	0.777 ±0.010 ↑	0.115 ± 0.003 (14.1%)	**6.908 ± 0.161 (15.0%)**
75%	Quantile estimation	QRDNN	0.836 ±0.017 ↑	0.150 ± 0.005	5.626 ± 0.089
	Direct PI estimation	QDDNN	0.814 ±0.009 ↑	**0.131 ± 0.004 (12.7%)**	**6.368 ± 0.141 (13.0%)**
		HN	0.824 ±0.013 ↑	0.135 ± 0.002 (9.9%)	6.238 ± 0.130 (10.7%)
80%	Quantile estimation	QRDNN	0.878 ±0.015 ↑	0.168 ± 0.006	5.282 ± 0.109
	Direct PI estimation	QDDNN	0.856 ±0.008 ↑	**0.149 ± 0.002 (11.1%)**	**5.862 ± 0.091 (11.0%)**
		HN	0.863 ±0.009 ↑	0.158 ± 0.002 (6.0%)	5.575 ± 0.053 (5.6%)
85%	Quantile estimation	QRDNN	0.913 ±0.013 ↑	0.191 ± 0.007	4.837 ± 0.132
	Direct PI estimation	QDDNN	0.893 ±0.008 ↑	**0.172 ± 0.003 (9.6%)**	**5.294 ± 0.099 (9.5%)**
		HN	0.895 ±0.010 ↑	0.182 ± 0.004 (4.7%)	5.014 ± 0.069 (3.7%)
90%	Quantile estimation	QRDNN	0.942 ±0.008 ↑	0.222 ± 0.008	4.299 ± 0.130
	Direct PI estimation	QDDNN	0.927 ±0.006 ↑	**0.204 ± 0.005 (8.0%)**	**4.644 ± 0.100 (8.0%)**
		HN	0.930 ±0.008 ↑	0.214 ± 0.005 (3.7%)	4.424 ± 0.094 (2.9%)
95%	Quantile estimation	QRDNN	0.967 ±0.007 ↑	0.272 ± 0.012	3.603 ± 0.137
	Direct PI estimation	QDDNN	0.958 ±0.003 ↑	**0.256 ± 0.005 (6.0%)**	**3.826 ± 0.067 (6.2%)**
		HN	0.964 ±0.007 ↑	0.266 ± 0.009 (2.3%)	3.679 ± 0.101 (2.1%)

Table 9
Mean PICP, AIW and Ratio results in Ciudad Real (solar energy).

PINC	Methodology	Model	PICP	AIW	Ratio
70%	Quantile estimation	QRDNN	0.768 ±0.009 ↑	0.171 ± 0.004	4.486 ± 0.067
	Direct PI estimation	QDDNN	0.719 ±0.009 ↑	0.144 ± 0.004 (16.2%)	5.012 ± 0.109 (11.7%)
		HN	0.703 ±0.019 ↑	**0.131 ± 0.007 (23.8%)**	**5.412 ± 0.153 (20.7%)**
75%	Quantile estimation	QRDNN	0.806 ±0.008 ↑	0.190 ± 0.005	4.238 ± 0.079
	Direct PI estimation	QDDNN	0.762 ±0.014 ↑	0.163 ± 0.004 (14.4%)	4.680 ± 0.093 (10.4%)
		HN	0.764 ±0.015 ↑	**0.157 ± 0.005 (17.3%)**	**4.869 ± 0.096 (14.9%)**
80%	Quantile estimation	QRDNN	0.841 ±0.008 ↑	0.211 ± 0.006	3.980 ± 0.091
	Direct PI estimation	QDDNN	0.808 ±0.011 ↑	**0.187 ± 0.004 (11.5%)**	4.322 ± 0.069 (8.6%)
		HN	0.831 ±0.016 ↑	0.193 ± 0.006 (8.9%)	**4.328 ± 0.076 (8.7%)**
85%	Quantile estimation	QRDNN	0.873 ±0.009 ↑	0.239 ± 0.006	3.655 ± 0.078
	Direct PI estimation	QDDNN	0.849 ±0.008 ↑	**0.218 ± 0.005 (8.7%)**	**3.894 ± 0.070 (6.5%)**
		HN	0.884 ±0.011 ↑	0.234 ± 0.010 (2.0%)	3.789 ± 0.128 (3.7%)
90%	Quantile estimation	QRDNN	0.906 ±0.005 ↑	0.275 ± 0.007	3.295 ± 0.078
	Direct PI estimation	QDDNN	0.895 ± 0.005	**0.259 ± 0.008 (5.8%)**	**3.460 ± 0.093 (5.0%)**
		HN	0.931 ±0.011 ↑	0.292 ± 0.015 (−6.0%)	3.205 ± 0.131 (−2.7%)
95%	Quantile estimation	QRDNN	0.942 ± 0.005	0.337 ± 0.009	2.802 ± 0.071
	Direct PI estimation	QDDNN	0.941 ± 0.005	**0.326 ± 0.008 (3.3%)**	**2.896 ± 0.062 (3.4%)**
		HN	0.964 ±0.006 ↑	0.357 ± 0.012 (−6.1%)	2.705 ± 0.079 (−3.4%)

in these cases, but as mentioned, it is at the cost of not reaching the target coverage. The other direct method (QDDNN) also fails to achieve the target coverage in a few cases (95% in Lugo, Table 6, and 90% and 95% in Ciudad Real, Table 9), but achieves it in all cases in Burgos (QRDNN fail to cover from 85% to 95% in Burgos, see Table 7).

A few issues deserve some further consideration. First, an explanation for why direct methods obtain narrower intervals in most cases. Both HN and QDDNN can do this in principle, because they directly consider width and coverage in the loss functions used to optimize them, whereas in quantile estimation (QRDNN) the focus is on the quantile loss and the posterior building of intervals. Therefore, QRDNN do not take interval width directly into account during the training process. Regarding the two direct methods, HN consider interval width and coverage directly, because these are the two goals of the multi-objective optimization. QDDNN also consider the two goals directly, because of the two terms in the quality-driven loss, one measures interval width and the other one penalizes intervals that do not reach the target coverage. Both terms are aggregated using parameter λ as a weight between width and coverage. Finally, in most of the cases where QRDNN obtain narrower intervals, it is because QRDNN fails to achieve the target coverage (Lugo and Ciudad Real 95%; Burgos 90% and 95%)

Second, it has been shown that HN always fulfill the target coverage. This may be due to the way solutions are selected from the Pareto front (Algorithm 2). The selection method used has a preference for the coverage of the PI to be above the target coverage (rather than below). HN use a validation set for this purpose, to construct a validation Pareto front. This way of selecting solutions, jointly to the fact that the validation set is expected to represent well the test set, makes HN able to achieve the coverage also in the test set. QRDNN on the other hand, does not consider interval width directly, and does not have a direct mechanism for satisfying the target coverage, so in some cases QRDNN may not achieve it. The other direct approach (QDDNN) also fails to satisfy the coverage in some cases. This may be caused by the QDDNN loss being an aggregation of two goals, interval width and the coverage penalty, and sometimes getting narrower intervals may overcome this penalty. HN on the other hand, once the Pareto front of solutions (all the tradeoffs between coverage and width) has been computed, it has a second stage for selecting the solution that satisfies the target coverage

Table 10
Statistical significance tests for QDDNN vs. QRDNN, HN vs. QRDNN and HN vs. QDDNN at different PINC values and regions.

PINC	Models	Lugo		Burgos		Córdoba		Ciudad Real	
		AIW	Ratio	AIW	Ratio	AIW	Ratio	AIW	Ratio
70%	QDDNN vs. QRDNN	+	+	+	+	+	+	+	+
	HN vs. QRDNN	+	+	+	+	+	+	+	+
	HN vs. QDDNN	=	+	+	+	=	=	+	+
75%	QDDNN vs. QRDNN	+	+	+	+	+	+	+	+
	HN vs. QRDNN	+	+	+	+	+	+	+	+
	HN vs. QDDNN	=	+	+	+	−	−	+	+
80%	QDDNN vs. QRDNN	+	+	+	+	+	+	+	+
	HN vs. QRDNN	+	+	+	+	+	+	+	+
	HN vs. QDDNN	=	+	+	+	−	−	−	=
85%	QDDNN vs. QRDNN	+	+	=	=	+	+	+	+
	HN vs. QRDNN	+	+	=	+	+	+	+	+
	HN vs. QDDNN	=	+	=	=	−	−	−	−
90%	QDDNN vs. QRDNN	=	=	−	=	+	+	+	+
	HN vs. QRDNN	=	+	−	=	+	+	−	−
	HN vs. QDDNN	=	+	=	=	−	−	−	−
95%	QDDNN vs. QRDNN	−	−	−	−	+	+	+	+
	HN vs. QRDNN	−	−	−	−	+	+	−	−
	HN vs. QDDNN	=	=	−	−	−	−	−	−

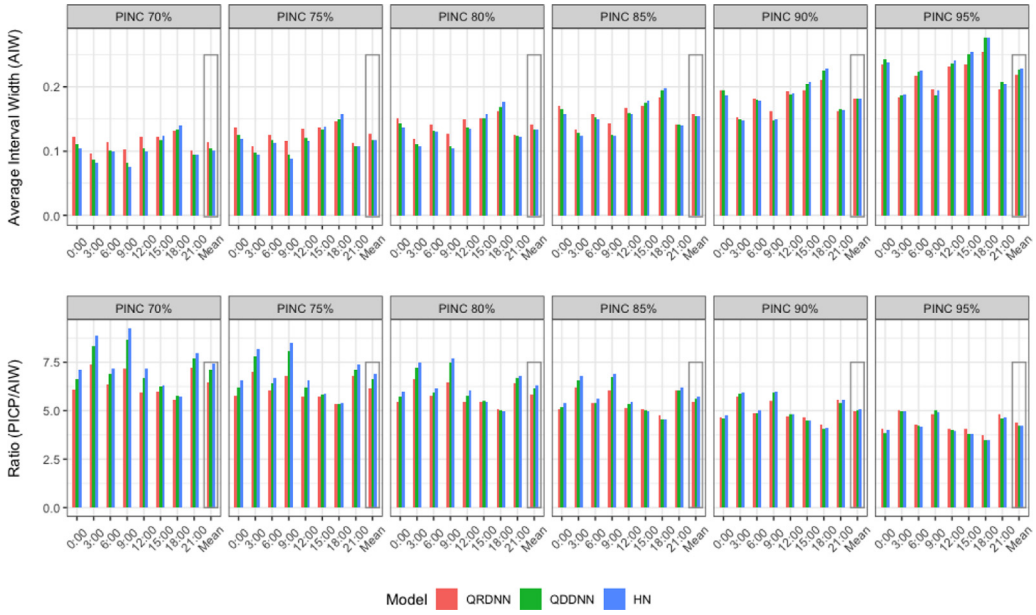


Fig. 7. AIW (up) and Ratio (down) in Lugo (wind energy) by time horizon.

out of the solutions in the Pareto front (disregarding the width at that stage, although both width and coverage were considered in the previous stage when the Pareto front was computed).

6.2. Time horizon analysis

To finish this section of results, the AIW and Ratio metrics will be broken down for every time horizon and shown graphically. As mentioned before, in wind energy regions, eight different horizons are predicted each day, from 00:00 to 21:00 in steps of 3 h. At the same time, results for solar energy are obtained at three horizons: 09:00, 12:00 and 15:00. It is important to remember that the horizon results are also averaged from ten different runs with ten different seeds.

Fig. 7 shows the value of the AIW (up) and Ratio (down) metrics for all temporal horizons and all target PINC in Lugo. Average values are displayed in the rightmost column. In relation to the width of the intervals, QDDNN and HN perform better than QRDNN most of the times. Only at 15:00 and 18:00 is the AIW in QRDNN smaller. In terms of the Ratio, direct estimation gets higher values of Ratio than QRDNN during the first half of the day, except for PINC 95%. At 15:00 and 18:00 the values are similar between the three different models.

In the case of Burgos (Fig. 8), direct PI estimation models (QDDNN and HN) also perform better than QRDNN at most of the times. Regarding the AIW (up), the three models are practically tied in their performance at 00:00, 18:00 and 21:00. However, in the rest of the horizons, QDDNN and HN clearly overtake QRDNN. As for the Ratio (down) metric, it can be seen how the direct PI estimation obtains larger values for every time horizon for PINC 70%, 75%, 80% and 85%. For the rest of the PINCs, the behavior is similar to the AIW, the direct estimation models perform better than quantile estimation, except between 6:00 and 12:00.

Next, the same results will be discussed for the solar energy regions. Firstly, the broken down AIW and Ratio in Córdoba are presented in Fig. 9. In terms of the AIW, for PINCs below 95%, results obtained by

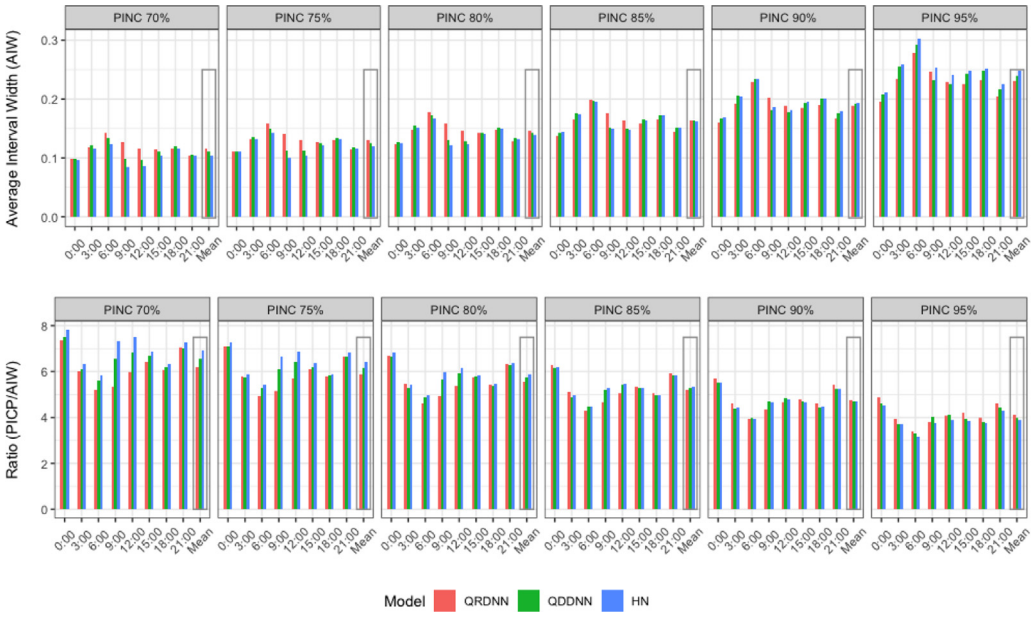


Fig. 8. AIW (up) and Ratio (down) in Burgos (wind energy) by time horizon.

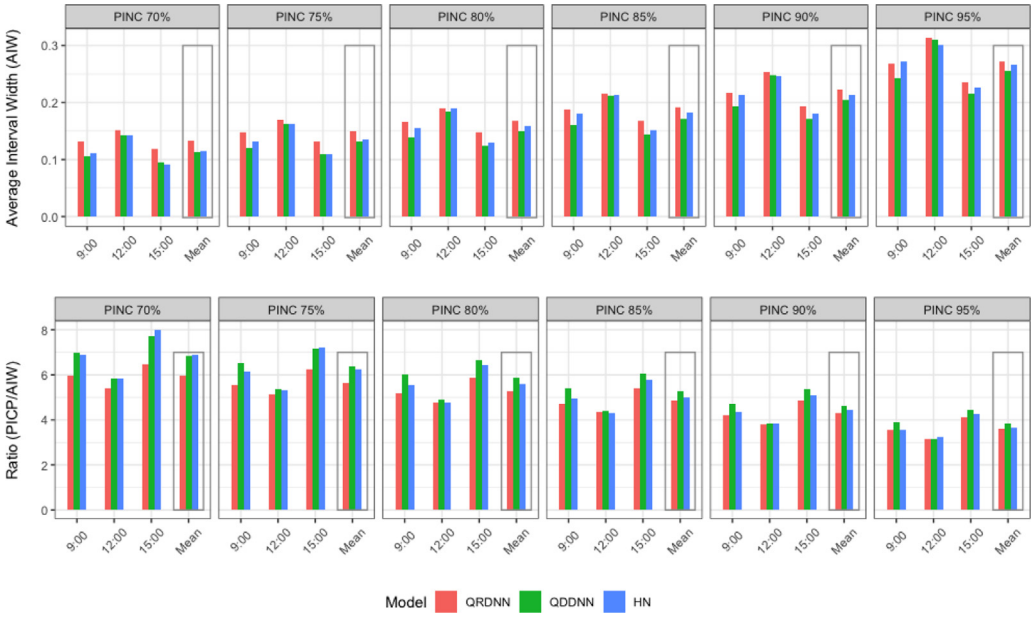


Fig. 9. AIW (up) and Ratio (down) in Córdoba (solar energy) by time horizon.

direct estimation models are better at 9:00 and 15:00. At 12:00, the value of the width is similar for all three models. In PINC 70% and 75%, both QDDNN and HN give a better Ratio than QRDNN for all three time horizons. From PINC 80%, Ratios for QDDNN are always the largest at 9:00 and 15:00, while the Ratio for HN starts to fail in comparison to QRDNN as the PINC grows.

Fig. 10 shows the results in the region of Ciudad Real (solar region). For PINC 70%, 75% and 80%, QRDNN is always the worst performing model for all the three time horizons in terms of both the AIW and the Ratio. For PINC 85%, direct estimation models outperform quantile estimation at 9:00 and 15:00, while at 12:00 the results are similar. For the remaining intervals, QDDNN gives the best values of AIW and Ratio at every time while HN deteriorates performance.

Finally, the time horizon metrics breakdown is summarized. For wind energy regions, both QDDNN and HN outperform QRDNN in AIW and coverage-width ratio for most of the horizons and PINCs. This does not happen only in central hours of the day and for the PINC 95%. In solar energy regions, QDDNN and HN generally obtain narrower PIs with bigger Ratio than QRDNN does for all time horizons and PINCs. Only for PINC 95% in Ciudad Real HN perform worse than both QDDNN and QRDNN, but is the only method achieving the required coverage.

7. Conclusions

In this work, two different DNN-based prediction interval estimation methodologies have been compared for renewable energy forecasting in a regional context. On the one hand, quantile estimation has been employed as a prior step to construct prediction intervals. On the other hand, direct estimation methods are able to directly output lower and upper bounds for PIs. Quantile Regression Deep Neural Networks (QRDNN) have been used as the main representative of the quantile

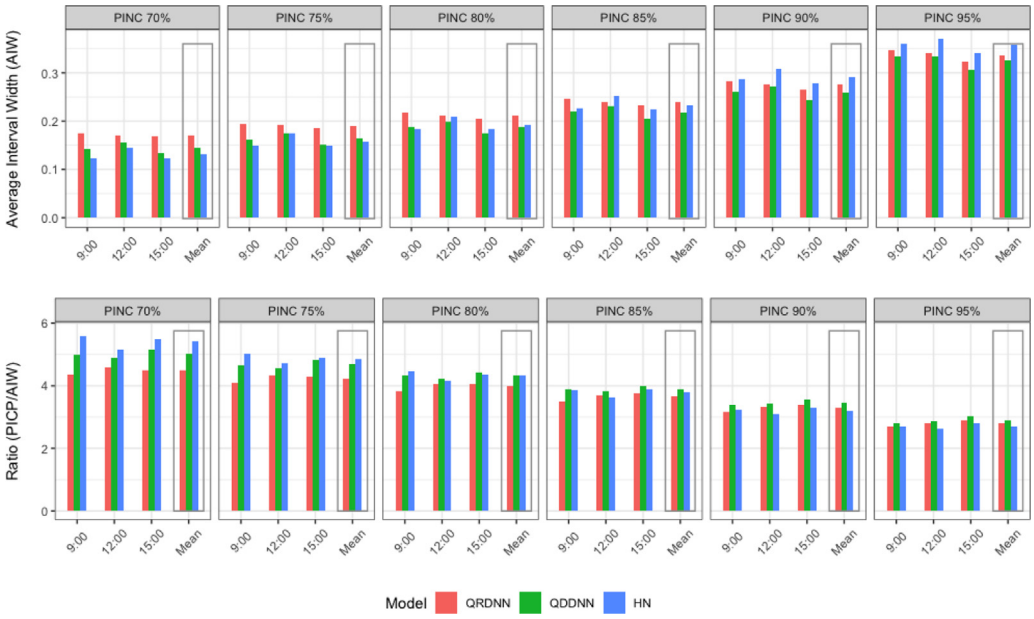


Fig. 10. AIW (up) and Ratio (down) in Ciudad Real (solar energy) by time horizon.

estimation methodology, being able to estimate conditional probability quantiles used for building centered PIs. Quality-Driven loss Deep Neural Networks (QDDNN) and Hypernetworks (HN) were used for direct estimation, optimizing width and coverage of PIs simultaneously.

These methods have been used to predict wind and solar energy production in four different Spanish regions (Lugo and Burgos for wind, Córdoba and Ciudad Real for solar). NWP variables were employed as inputs to the models, together with time-related variables taking advantage of embeddings. In wind energy regions, probabilistic predictions were made for all time horizons available (00:00, 03:00, ..., 21:00), whereas for solar regions predictions were limited to sunshine hours (09:00, 12:00 and 15:00). Trained models were used to predict 6 different PIs, with required coverages of 70%, 75%, 80%, 85%, 90% and 95%.

In the aggregated results (average across all time horizons), direct PI models (HN and QDDNN) clearly outperform quantile estimation in terms of width and coverage-width ratio for the 70%–80% in all regions, and also 85%, 90%, 95%, depending on the region. Direct estimation is able to improve these metrics compared to QRDNN, up to more than 20% in some cases, depending on the target PINC and the region. This may be due to the direct consideration of PI width and coverage in the losses optimized by direct methods. In most of the cases where QRDNN obtain narrower intervals, it is because QRDNN fails to achieve the target coverage (Lugo and Ciudad Real 95%; Burgos 90% and 95%)

Although all models do a correct job of achieving the target coverage in most cases, HN are the only ones that do so for every target coverage in every region, while QRDNN and QDDNN fail in some cases for large target coverages. This good behavior of HN may be the result of the procedure used to select the optimal point in the validation Pareto front, which has a preference for obtaining solutions that achieve the required coverage.

Disaggregating the AIW and Ratio metrics by time horizon, QDDNN and HN models also perform better than QRDNN in the majority of the studied hours with no differences between regions.

Regarding the two direct methods, QDDNN and HN, while both of them work well in terms of PI width and Ratio, the novel formulation of direct PI estimation as a multi-objective problem using hypernetworks has resulted in a method that allows to obtain PIs for every possible target coverage, in contrast to QDDNN, which has to be trained for a set of predefined target coverages. HN also eliminate the need to define the coverage-width trade-off parameter λ required by QDDNN.

Finally, future research could focus on adapting direct PI estimation methods, both QD-loss and HN, to other neural architectures such as Long Short-Term Memory Neural Networks (LSTM) or Recurrent Neural Networks (RNN), that deal with sequences of data with time dependencies. This framework could be useful when modeling renewable energy uncertainty in a context where, for instance, meteorological information is not available, and predictions rely on historical data.

CRediT authorship contribution statement

Antonio Alcántara: Investigation, Software, Data curation, Validation, Writing – original draft. **Inés M. Galván:** Conceptualization, Investigation, Methodology, Funding acquisition, Investigation, Supervision, Writing - review. **Ricardo Aler:** Conceptualization, Investigation, Methodology, Software, Funding acquisition, Investigation, Supervision, Writing - review.

Declaration of competing interest

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

Acknowledgments

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 for APC: Universidad Carlos III de Madrid (Read & Publish Agreement CRUE-CSIC 2022)

References

Aler, R., Huertas-Tato, J., Valls, J.M., Galván, I.M., 2019. Improving prediction intervals using measured solar power with a multi-objective approach. Energies 12 (4713).

Andrade, J.R., Bessa, R.J., 2017. Improving renewable energy forecasting with a grid of numerical weather predictions. IEEE Trans. Sustain. Energy 8, 1571–1580.

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.

Bessa, R.J., 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 (1402).

Cannon, A.J., 2018. Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes. Stochastic Environ. Res. Risk Assess. 32, 3207–3225.

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.

David, M., Luis, M.A., Lauret, P., 2018. Comparison of intraday probabilistic forecasting of solar irradiance using only endogenous data. Int. J. Forecast. 34, 529–547.

Galván, I.M., Huertas-Tato, J., Rodríguez-Benítez, F.J., 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.

Galván, I.M., Valls, J.M., Cervantes, A., Aler, R., 2017. Multi-objective evolutionary optimization of prediction intervals for solar energy forecasting with neural networks. Inform. Sci. 418, 363–382.

Guo, C., Berkhahn, F., 2016. Entity embeddings of categorical variables. arXiv preprint arXiv:1604.06737.

Ha, D., Dai, A., Le, Q.V., 2016. Hypernetworks. arXiv preprint arXiv:1609.09106.

Hatalis, K., Lamadrid, A.J., Scheinberg, K., Kishore, S., 2017. Smooth pinball neural network for probabilistic forecasting of wind power. arXiv preprint arXiv:1710.01720.

He, Y., Li, H., 2018. Probability density forecasting of wind power using quantile regression neural network and kernel density estimation. Energy Convers. Manage. 164, 374–384.

Hu, Y., Zhan, W., Tomizuka, M., 2018. Probabilistic prediction of vehicle semantic intention and motion. In: 2018 IEEE Intelligent Vehicles Symposium. IV, IEEE pp. 307–313.

Khosravi, A., Nahavandi, S., 2013. Combined nonparametric prediction intervals for wind power generation. IEEE Trans. Sustain. Energy 4, 849–856.

Khosravi, A., Nahavandi, S., Creighton, D., Atiya, A.F., 2010. Lower upper bound estimation method for construction of neural network-based prediction intervals. IEEE Trans. Neural Netw. 22, 337–346.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Li, C., Tang, G., Xue, X., Chen, X., Wang, R., Zhang, C., 2020. The short-term interval prediction of wind power using the deep learning model with gradient descend optimization. Renew. Energy 155, 197–211.

Li, P., Zhang, C., Long, H., 2019. Solar power interval prediction via lower and upper bound estimation with a new model initialization approach. Energies 12 (4146).

Lian, C., Chen, C.P., Zeng, Z., Yao, W., Tang, H., 2016. Prediction intervals for landslide displacement based on switched neural networks. IEEE Trans. Reliab. 65, 1483–1495.

Liu, F., Li, C., Xu, Y., Tang, G., Xie, Y., 2021. A new lower and upper bound estimation model using gradient descend training method for wind speed interval prediction. Wind Energy 24, 290–304.

Martin, R., Aler, R., Valls, J.M., Galván, I.M., 2016. Machine learning techniques for daily solar energy prediction and interpolation using numerical weather models. Concurr. Comput.: Pract. Exper. 28, 1261–1274.

Navon, A., Shamsian, A., Chechik, G., Fetaya, E., 2020. Learning the pareto front with hypernetworks. arXiv preprint arXiv:2010.04104.

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., dÁlché Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems,Vol. 32. Curran Associates, Inc. pp. 8024–8035.

Pearce, T., Brintrup, A., Zaki, M., Neely, A., 2018. High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. In: International Conference on Machine Learning. PMLR, pp. 4075–4084.

Ruchte, M., 2021. Cosmos - efficient multi-objective optimization for deep learning. https://github.com/ruchtem/cosmos.

Torres-Barrán, A., Alonso, Á., Dorronsoro, J.R., 2019. Regression tree ensembles for wind energy and solar radiation prediction. Neurocomputing 326, 151–160.

Wan, C., Xu, Z., Pinson, P., 2013a. Direct interval forecasting of wind power. IEEE Trans. Power Syst. 28, 4877–4878.

Wan, C., Xu, Z., Pinson, P., Dong, Z.Y., Wong, K.P., 2013b. Optimal prediction intervals of wind power generation. IEEE Trans. Power Syst. 29, 1166–1174.

Wilcoxon, F., 1992. Individual comparisons by ranking methods. In: Breakthroughs in Statistics. Springer, pp. 196–202.

Wu, W., Chen, K., Qiao, Y., Lu, Z., 2016. Probabilistic short-term wind power forecasting based on deep neural networks. In: 2016 International Conference on Probabilistic Methods Applied to Power Systems. PMAPS, IEEE, pp. 1–8.

Zitzler, E., Thiele, L., 1998. Multiobjective optimization using evolutionary algorithms—a comparative case study. In: International Conference on Parallel Problem Solving from Nature. Springer, pp. 292–301.