Day-Ahead Parametric Probabilistic Forecasting of Wind and Solar Power Generation using Bounded Probability Distributions and Hybrid Neural Networks

Day-Ahead Parametric Probabilistic Forecasting of Wind and Solar Power Generation using Bounded Probability Distributions and Hybrid Neural Networks

Theodoros Konstantinou, *Member, IEEE* and Nikos Hatziargyriou, *Life Fellow, IEEE*

***Abstract*—The penetration of renewable energy sources in modern power systems increases at an impressive rate. Due to their intermittent and uncertain nature, it is important to forecast their generation including its uncertainty. In this paper, an ensemble artificial neural network is applied for day ahead solar and wind power generation parametric probabilistic forecasting. The proposed architecture includes two components: a sub-models component and a Meta-Learner component. The first component includes an ensemble of artificial neural networks that have the ability to estimate the parameters of an underlying probability distribution. The Meta-Learner is responsible for grouping the training samples based on the estimated level of generation, through a classification-clustering process and use the output of the corresponding sub-models to calculate the final parametric probabilistic estimation. The proposed model is compared to both parametric and non-parametric state of the art probabilistic techniques for solar and wind power generation forecasting, exhibiting superior performance.**

***Index Terms*—Artificial neural networks, ensemble forecasting, parametric probabilistic forecasting, probability density estimation**

NOMENCLATURE

α_f^j, b_f^j	Kumaraswamy parameters for the final prediction of j_{th} training sample
α_i^j, b_i^j	Kumaraswamy parameters of the i_{th} submodel for j_{th} training sample
c_r^j	Clustering participation weight of the j_{th} train sample in the r_{th} cluster
F^j	Estimated cumulative distribution function of j_{th} training sample
F_K	Cumulative distribution function of Kumaraswamy distribution
F_K^{-1}	Quantile function of Kumaraswamy distribution
f_K	Probability density function of Kumaraswamy distribution
J	Multi-objective loss function
L	Numerical quadrature approximation of CRPS
L_{CE}^j	Categorical cross-entropy of the j_{th} train sample
N	Number of training samples
N_C	Number of clusters
N_{epochs}	Number of epochs/training iterations
t_r^j	r_{th} element in the j_{th} cluster vector
$w_{\alpha,i}^t$	Synaptic weights of parameter α node of i_{th} cluster in epoch t
$w_{b,i}^t$	Synaptic weights of parameter b node of i_{th} cluster in epoch t
$\overline{y}$	Upper boundary of the estimated random variable
$\underline{y}$	Lower boundary of the estimated random variable
y^j	Observed target value of j_{th} training sample
λ	Training learning rate
$\mathbb{1}$	Heaviside function
k	Transition parameter of the logistic function
n	Number of sub-intervals in numerical quadrature integration
p	Probability used in cumulative distribution function

T. Konstantinou is with the School of Electrical and Computer Engineering, National Technical University of Athens, Greece, e-mail: tkonstantinou@mail.ntua.gr.

N. Hatziargyriou is with the School of Electrical and Computer Engineering, National Technical University of Athens, Greece and University of Vaasa, Finland, e-mail: nh@power.ece.ntua.gr

I. INTRODUCTION

A. Motivation

In recent decades, power systems are characterized by high penetration of renewable energy generation. Renewable energy sources' (RES) generation is highly dependent on weather conditions, making it uncertain and intermittent by nature. These uncertainties create challenges for system operators in optimally scheduling the operation of their systems and in ensuring their secure and reliable operation [1]–[3]. Furthermore, the uncertainty of RES affects trading in energy markets, as RES producers might be subject to penalties if they fail to meet their offers in day ahead markets [4]. In order to cope with these challenges, the intermittent and unpredictable nature of RES must be properly taken into account through uncertainty analysis and probabilistic forecasting processes [5]. Stochastic optimization techniques offer a number of advantages over deterministic ones, when applied for system operational scheduling. The main reason is that perfect deterministic forecasts are not possible in power systems with high penetration of RES generation [6].

Classical approaches for univariate probabilistic RES generation forecasting, are parametric approaches, that estimate probability densities that rely on assumptions for the underlying distribution of future RES generation, whose parameters are estimated through statistical and machine learning methods. For instance, the most commonly assumed distributions are the Gaussian, Beta, Generalized Logit-Normal and Weibull [7]. Although it is convenient to develop models based on such assumptions, the distribution of wind power or solar power must be selected carefully in order for the

assumption to be valid. This is undoubtedly a drawback of parametric approaches that motivated many researchers to look for distribution-free density approaches, i.e., approaches that do not rely on a specific assumption for the densities to model. Certainly the most popular distribution-free approach, also referred to as non-parametric, is quantile regression [8], which allows to relax the use of distributional assumptions for the case of univariate probabilistic forecasting. It has achieved great success in the Global Energy Forecasting Competition 2014 (GEFCom 2014) in both solar and wind power forecasting task, becoming a mainstream solution due to its state-of-the-art performance and simplicity of use. However, it requires parallel models to be fitted for each quantile, which raises the cost of computation when the whole distribution is needed. In addition, it only provides discrete quantiles, which may lead to quantile crossing. Additionally, the uncertainty information derived from these results may not cover the entirety of applications used in power systems where stochastic optimization is used and probability densities are preferred [9]. Another mainstream approach for distribution-free models, is the mixture density method, which is able to model entire distributions by adding a finite number of kernels. This method may overcome the drawbacks mentioned, but provides the final estimation in a finite summation of kernels which might be difficult to efficiently incorporate in stochastic optimization problems, e.g., applications that require the inversion of a cumulative distribution function (CDF) [4], [10]. Eventually, it remains an open issue to develop an efficient and continuous probabilistic forecasting model that obtains an easily applicable distribution without the drawbacks of assuming a distribution.

B. Related Work

Extensive research has been made on the probabilistic estimation of solar and wind power generation at multiple time step horizons ahead. Probabilistic forecasting methods can be classified into three categories based on the way they are modeling the uncertainty of predictions: quantile prediction or prediction intervals (PI), density estimations and ensemble predictions.

Quantiles and PI estimation methods are based on quantile regresion, a process which minimizes the quantile error. These approaches extend from the simple quantile linear regression to more advanced methods such as quantile random forests, gradient boost machines and neural networks. There is an extensive literature on quantiles and PI estimation methods applied in RES generation forecasting [11]–[15]. The main idea of these works is to construct prediction intervals or predict a set of quantiles with predefined nominal probabilities. In most cases, PI are easily constructed by combining a pair of quantiles based on the the desired confidence level. However, quantiles, and in extend PI, provide only partial information of the outcome's uncertainty. This shortcoming of these methods, may be negligible in some applications of power systems, but is important on stochastic programming where the probability distribution of future wind or solar power generation is often required. In order to achieve this level of details using PI estimation models either a large number of such models must by constructed or either using numerical methods to estimate the distribution from which the quantiles were drawn, thus transforming the problem back to a density estimation problem. Another drawback of these methods, compared to the proposed method is that these models are optimized on a certain level of probability, without considering the goodness-of-fit of their predictions to the actual distribution of the data.

Advances have been made in probability distribution estimation, in order to obtain full uncertainty information, through mixture models. Mixture models are probabilistic models that correspond to mixture distributions, created as a linear combination of a finite number of distributions of a sub-population of the data. A very popular mixture model is the kernel density estimation (KDE) [16]–[18]. The most popular approach of KDE, for RES generation forecasting, estimates the probability density of a prediction based on a finite population of historical data, selected by a distance metric, like the k-nearest neighbors method as proposed in [17]. Even if the KDE is a powerful tool as a universal density approximation, it is still limited by the finite population selection which is not adaptive and suffers from the curse of dimensionality. The drawbacks of KDE were addressed by the Mixture Density Network (MDN). MDN is a neural network model that was proposed by Bishop [19] and can model the conditional probability distribution of the target data using mixture distributions within the neural network. MDNs have been proposed in recent years for probabilistic forecasting of RES generation, based on their major advantage which is that they are distribution-free methods. In [20], the authors propose the Improved Deep Mixture Density Network (IDMDN) in which the kernels of the network are modelled using Beta kernels. In contrast to the common Gaussian kernels, the Beta kernels are defined on a range of $[0, 1]$, addressing the density leakage problem. In another recent work on distribution-free methods, the authors study the use of a Conditional normalizing flow model to learn the mapping from historical weather and wind power data to a joint probability distribution over future wind power outputs [21]. Another popular approach of probabilistic forecasting for RES generation is the Gaussian Process (GP) [22]–[24]. GP is a regression model that fits a multi-variate Gaussian distribution in the training data and models the distribution of functions that can describe these data. The main disadvantages of the GP method is that it is time-consuming due to its covariance matrix inversion that limits its performance in datasets with a large number of train samples. Nevertheless, GP is a very popular approach to probabilistic forecasting.

In addition, a popular and fairly simple method for probabilistic forecasting is a resampling process through bootstrapping. In [25], a very short term parametric probabilistic forecasting technique is applied to solar power forecasting. This technique uses extreme learning machine applied in a bootstrap procedure to create the Bootstrap-Extreme Learning Machines (BELM). In [26], similarly to [25], the authors propose the use of Extreme Learning Machines and several bootstrap methods, for predicting probability intervals, after estimating the parameters of a Gaussian probability distribution. Even though both of these methods construct prediction

intervals, the boundaries of the intervals are estimated by assuming the quantile function of a Gaussian distribution. The distribution is defined by the expected value and variance of an ensemble of point forecasts, making this method partially parametric.

C. Main Contributions

As a basis for this work, we get inspiration from [27], which proposed the use of probability density estimation neural network (PDE-NN) for estimating the probability distribution of wind speed, observed in a wind farm, based on numerical weather predictions. PDE-NN can be described as a MDN variation, where instead of a mixture of distributions, the output is given by a set of parameters which define a probability distribution. One of the main contributions of this work, was the proposal of the Kumaraswamy distribution as an efficient candidate for distribution assumption in wind speed probabilistic forecasting. Within this work, we extend the evaluation of the Kumaraswamy distribution in RES generation forecasting based on its two main advantages: it is a bounded distribution and it is one of the simplest two-parameter distributions that can alter its shape in order to model a variety of distributions, making it a perfect candidate for assumed distribution selection, both for solar and wind power generation.

Additionally, the PDE-NN, is introduced in an ensemble based model, where multiple PDE-NN sub-models are trained in order for the final estimation to be made. One major difference between the MDN and our proposed methodology is in the creation of the final estimation. As mentioned in the previous section, MDN create a mixture of distributions in a linear combination manner. This can introduce difficulties in applications that require usage of the inverse of cumulative distribution functions or in scenario generation methods. A main advantage of the Kumaraswamy distribution and the proposed methodology is that it does not predict the final distribution of the target data as a linear combination of distributions but it estimates the parameters of the final Kumaraswamy distribution as a linear combination of the parameters from the multiple probabilistic sub-models. This provides the ability of an easily invertible distribution function. More information about the use of Kumaraswamy distribution is provided later in the paper.

The linear combination of parameters, is produced by a component called Meta-Learner, which is responsible for estimating the weights with which the parameters are multiplied before the linear combination. Within this work, we introduce the idea of training each sub-model in a specific cluster of similar train samples. Therefore, we propose that the participation weights of parameters to be related to the clustering process in order to represent the probability of each train sample to belong to every cluster. Based on this idea, the clustering process is held within the training process of the entire model and is optimized in parallel with the training of the sub-models and the Meta-Learner. The innovation of the clustering idea, is the proposal of a supervised clustering process within the model, that clusters the training samples based on the classification of the estimated outcome instead of the similarity of input features. Using this supervised clustering process, each sub-model's training is focused on a cluster of the training samples with relatively close level of power generation. This process creates sub-models that have high accuracy on their specified level of generation.

In addition to the previously mentioned advantages, based on the flexibility of the Kumaraswamy distribution to model several distributions' shapes inside a bounded domain and the extensive use of artificial neural networks for this task, the proposed methodology is tested both for solar and wind power generation predictions. It is apparent that these tasks depend on very different input features, but our assumption is that the presented method is general and agile enough, in order to be able to model both these two prediction tasks. Moreover, the main component of the proposed model is based on the idea of an ensemble of probabilistic models designed as PDE-NNs as proposed in [27]. The main advantage of PDE-NNs is that they can incorporate most NNs architectures. This makes it possible for several known methods, used in the literature, to be incorporated in order to increase their performance and be transformed into probabilistic methods (e.g. convolutional neural networks, recurrent neural networks, etc.). In addition, the model is compatible with most of state-of- the-art pre-processing methods, e.g. signal decomposition in the temporal domain, which have proven to increase the performance of forecasting models, such as the recently proposed ICEEDMAN framework [28], [29].

The remainder of this paper is organized as follows: Section II presents the overall architecture of the proposed model and Section III describes in detail its components. Section IV provides all the information of the evaluation test cases, assessment metrics and benchmark models. In Section V, the evaluation results of the proposed model are presented and discussed. Finally, in Section VI, the conclusions of this work are summarized.

II. Proposed Method

The proposed model is based on the use of multiple parametric probabilistic sub-models in order to create an ensemble of probabilistic forecasts, and estimate a final probabilistic forecast. The suggested architecture, shown in Figure 1, consists of the following components: The Sub-models component (a) and the Meta-Learner component (b), which for the rest of this paper will be simply referred to as Meta-Learner. Since the Kumaraswamy distribution is selected, the distribution parameters displayed in brackets, in Figure1, correspond to the distribution parameters defining a Kumaraswamy distribution $[\alpha, b]$. The arrows in Figure 1 and data in brackets display the flow of information within the proposed model. All components are based on artificial neural networks, participate as different branches of a single network and are trained simultaneously in a single training process.

The Sub-models component consists of a number N_C of neural network-based sub-models. As shown in Figure 1 (a), each sub-model acts as a different independent branch of the same neural network which produces parametric probabilistic

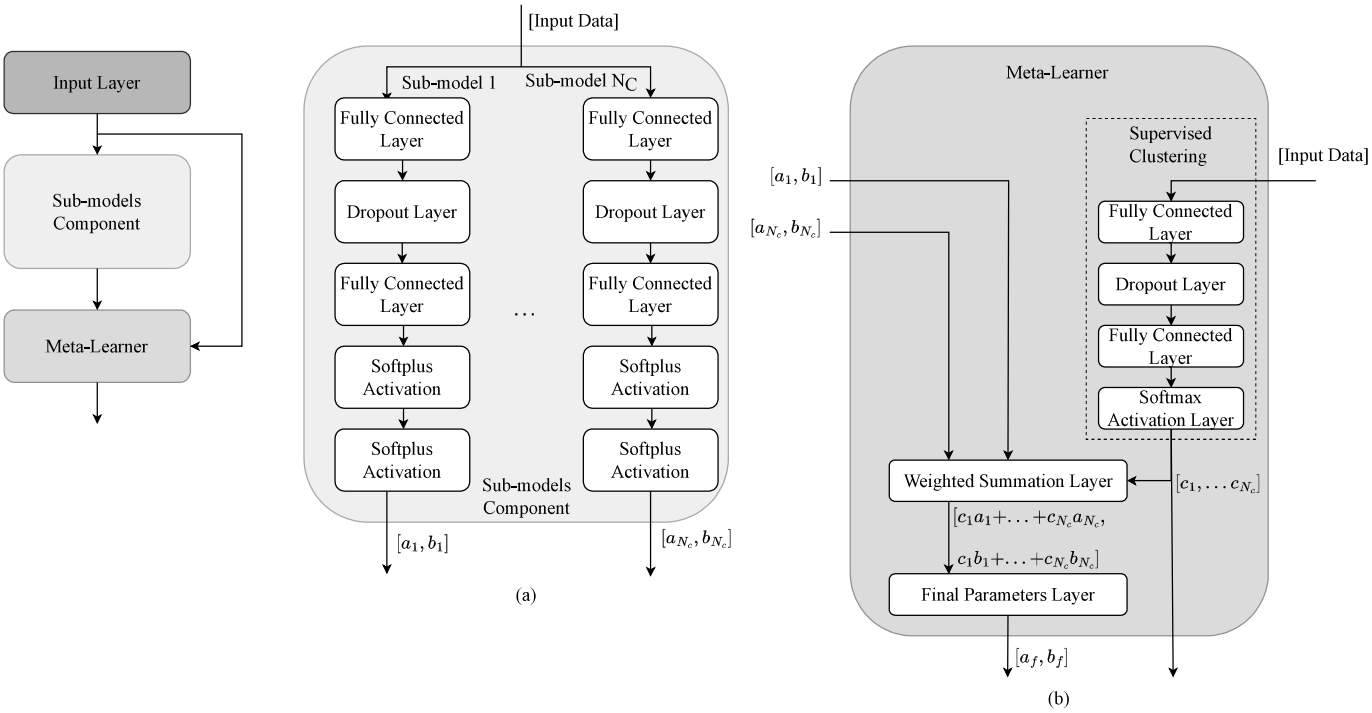


Fig. 1: Overall architecture of the proposed model

forecasts. The goal of this component is to present to each sub-model the same input vector and for each sub-model to produce a different output in the form of a probability density function's (PDF) parameters. The difference in outputs of each sub-model is achieved in two ways. Firstly, the synaptic weights are randomly initialized with samples from a continuous distribution. Additionally, since non-linear activation functions are used inside the sub-models, even the slightest difference in the initial values will provide different outcomes. Secondly, as it will be explained in more detail further in the paper, the synaptic weights of each branch are updated based on their cluster participation in the final prediction and its error. This participation is calculated based on the cluster each training sample is assigned to. Through the training process, the back-propagation of loss is weighted by the participation in each cluster. Therefore, each sub-model is optimized on the training samples of its assigned cluster. This leads to sub-models with different synaptic weights and different outputs although they are provided with the same input features. These outputs are passed to the Meta-Learner and are used as the input data for the final parametric probabilistic estimation.

The use of clustering in forecasting is a well established process, as it aims to divide the training data in groups, assuming that similar training data might have the same behaviour, thus making the forecasting procedure more accurate. In the literature, clustering is mostly applied in the input features of the training samples, assuming that similar input features will give similar outputs [30]–[33]. This approach is highly sensitive to noise and errors from the numerical weather predictions used as input in most cases. This leads to a case where the probability distribution of possible power generation for each cluster of data is distributed in the entire values' domain. This may lead to sub-models with lower accuracy, since they were trained on similar input data but were expected to estimate dissimilar outputs. In order to overcome this sensitivity of clustering on noise and errors present in the input features, a clustering process based on the value of generation is proposed. The train samples are assigned into N_C clusters based on the target value (value of generation) and the information about to which cluster each sample belongs to is transformed into classification labels through one-hot encoding. Therefore, the input features are not assigned to clusters based on their distance from each other but they are sorted and divided into equally spaced N_C sub-domains of the power generation's domain. The number of clusters N_C is considered a hyper-parameter of the model and its optimal value can be determined through a hyper-parameter optimization method. Within this work, the optimal N_C is selected in each case study separately, through grid-search. The Meta-Learner is the model's component responsible for combining all the information from the sub-models and the supervised clustering component in order to make the final estimation. This is achieved by calculating the distribution's parameters of the final estimation as the weighted linear combination of the distributions' parameters from all sub-models. The forward pass of information in the proposed model through the training phase is summarised in Algorithm 1. The following two sections of the paper present a detailed analysis of the processes of both components.

III. Sub-models Component

A. Probability Density Estimation Neural Networks

PDE-NNs are a unique type of artificial neural networks, capable of estimating probability density functions. The idea

Algorithm 1 Forward pass of information in the proposed model through the training phase

Initialize synaptic weights of all layers
$e \leftarrow 1$
while $e \leq N_{epochs}$ **do**
  Estimate $[\alpha_1^j, b_1^j], ..., [\alpha_{N_C}^j, b_{N_C}^j] \quad \forall j \in N$
  Estimate $[c_1^j, ..., c_{N_C}^j] \quad \forall j \in N, \quad s.t \quad \sum_{i=1}^{N_C} c_i^j = 1$
  Estimate $a_f^j = \sum_{i=1}^{N_C} c_i^j a_i^j : \forall j \in N$
  Estimate $b_f^j = \sum_{i=1}^{N_C} c_i^j b_i^j : \forall j \in N$
  $e \leftarrow e + 1$

behind PDE-NN is the exploitation of artificial neural networks (ANN) as a universal approximator. Through the training process, the PDE-NN calculates the continuous PDF of every target value by estimating its parameters. In order to model the parameters vector, its output layer is replaced with a parameters layer. This layer has as many nodes as the number of the parameters that define the selected distribution. These parameters are estimated as a non-linear combination of the PDE-NN's output values from the last hidden layer. The main difference between a common output layer and the parameters' layer is that the common output layer estimates the required output variable, while the parameters' layer estimates only the parameters of the output's underlying distribution. This gives the advantage to the PDE-NN, using the parameters, not only to estimate the expected value of the distribution as a deterministic forecast, but also estimate quantiles, prediction intervals or create scenarios of RES generation.

B. Kumaraswamy Distribution

The last step in building a PDE-NN is the decision of the probability distribution to be fitted. This decision must consider all the physical constraints of the variable we are predicting. RES generation as described, is a bounded variable between zero and the installed capacity of the RES plant. As studied in [27], a well fitted solution is the Kumaraswamy distribution defined by two non-negative shape parameters, a and b. In its simplest form, the distribution has a support of $(0, 1)$. In a more general form, the normalized random variable y is replaced with an un-shifted and un-scaled random variable, which is then scaled and shifted by the chosen boundaries $[\underline{y}, \overline{y}] \subseteq \mathbb{R}$. Using this transformation, the distribution can be stretched in order to include the boundaries of 0 and 1. The Kumaraswamy's PDF and CDF are shown in (1) and (2), respectively [34].

$$f_K(y) = ab[\frac{y-\underline{y}}{\overline{y}-\underline{y}}]^{a-1}(1-[\frac{y-\underline{y}}{\overline{y}-\underline{y}}]^a)^{b-1} \quad , y \in (\underline{y}, \overline{y}) \quad (1)$$

$$F_K(y) = 1-(1-[\frac{y-\underline{y}}{\overline{y}-\underline{y}}]^a)^b \quad , y \in (\underline{y}, \overline{y}) \quad (2)$$

The PDF and CDF are much simpler than commonly used probability distributions in probabilistic forecasting, such as Beta, Normal, Logit-normal and Weibulll. Furthermore, to the authors' knowledge, the Kumaraswamy distribution, is the only two-parameter quantile family bounded in a specific

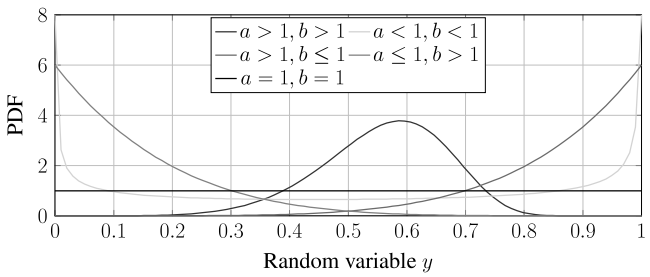


Fig. 2: Kumaraswamy PDF for different values of shape parameters a and b

space so simply defined. The CDF is readily invertible to yield the quantile function as shown in (3), which facilitates ready quantile-based modeling.

$$F_K^{-1}(p) = (\overline{y}-\underline{y})(1-(1-p)^{\frac{1}{b}})^{\frac{1}{a}} + \underline{y} \quad , p \in [0, 1] \quad (3)$$

In Figure 2, the probability density functions of the normalized Kumaraswamy distribution for different values of parameters a and b are displayed. As observed, the Kumaraswamy distribution has the basic shape properties of one of the most commonly used bounded distributions, the Beta distribution. Thus, by altering the shape parameters, we are able to change the shape of the PDF. This is exactly the goal of the training process of the PDE-NN sub-models, namely to discover the pairs of parameters that for each input vector minimize the CRPS for observing the target value.

C. Objective function

The training process of an ANN is based on the backward propagation of a loss and the update of the ANN's weights to minimize this loss. Therefore, the training process can be described as a minimization problem having an objective function dependent on the error between the estimated and target value. The idea behind PDE-NN is to use a loss function that considers the error of an entire probability distribution. The most common performance measures of probabilistic forecasts of a scalar observation is the Continuous Ranked Probability Score (CRPS) and the maximum likelihood estimation (MLE). According to [35], CRPS is more robust to distributional misspecification in comparison to the maximum likelihood estimation, making it a better choice as the objective function in this work. To further establish the superiority of the CRPS as a loss function, in comparison to the MLE, both these loss functions will be evaluated in the context of this paper. CRPS is a quadratic measure of the difference between the estimated cumulative distribution function F and the empirical cumulative distribution function of the observation as shown in (4). In this context, the empirical cumulative distribution function is modeled by the Heaviside function $\mathbb{1}$ (5) .

$$CRPS(F^j, y^j) = \int_{\mathbb{R}} [F^j(z) - \mathbb{1}(z \geq y^j)]^2 dz \quad : \forall j \in N \quad (4)$$

$$\mathbb{1}(z \geq y^j) = \begin{cases} 1, & \text{if } z \geq y^j. \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

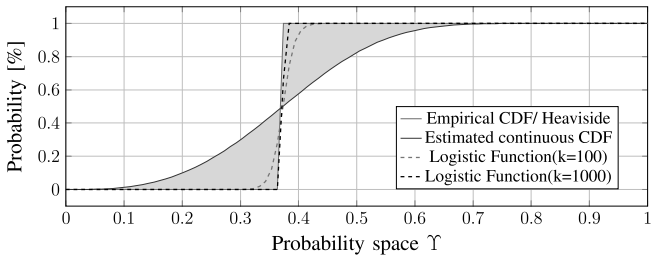


Fig. 3: Different approximations of CDF

where z is a dummy variable for integration. In order for the CRPS to be introduced as a differentiable objective function, for the training process, the Heaviside function is replaced by the differentiable logistic function as a smooth approximation (6).

$$\mathbb{1}(z \geq y^j) \approx \frac{1}{1+e^{-k(z-y^j)}} \quad : k \geq 0, \forall j \in N \tag{6}$$

where a larger k corresponds to a sharper transition at the point $z = y$. For the purpose of training a probabilistic model using the CRPS as the objective function, a value of $k = 10^3$ is proposed. An example of the different approximations of CDF are displayed in Figure 3, including two logistic approximations with $k = 10^2$ and $k = 10^3$. By minimizing the CRPS, one minimizes the difference between the estimated and empirical CDF. Therefore, a closer approximation to the empirical CDF will lead to a better estimation of the CRPS to be minimized. Finally, in order to reduce the computational time of the training process, the CRPS integral is approximated using numerical quadrature with $n = 10$, as shown in (7)

$$\int_{z_0}^{z_1} g(z)dz \approx \frac{z_1 - z_0}{n}\left(\frac{g(z_0)}{2} + \sum_{i=1}^{n-1} g(z_0 + i\frac{z_1 - z_0}{n}) + \frac{g(z_1)}{2}\right) \tag{7}$$

$$g(z) = [F^j(z) - \frac{1}{1+e^{-k(z-y^j)}}]^2 \quad : \forall j \in N \tag{8}$$

Since the Kumaraswamy distribution is selected, the final architecture of the PDE-NN sub-models includes a two parameters output layer. In order to constraint the Kumaraswamy distribution's parameters to be greater than 0, the softplus activation function, prior to the parameters' layer is used. Softplus is a commonly used activation function, that is used to constrain the output of a layer to always be positive.

IV. Meta-Learner

A. Supervised Clustering

The short-coming of this method is that in real-time operation of the model, we do not have prior knowledge of the magnitude of power generation, in order to decide which sub-models should participate in the prediction. In order to overcome this, a supervised clustering process based on classification is proposed. If by using the input features we are able to determine in which cluster the actual power generation will be observed, then the clustering problem is transformed into a classification problem and can be addressed. To achieve this, the Supervised Clustering component is proposed which is responsible for predicting, for each training sample, in which cluster belongs to. Prior to the training process, the train samples are sorted into clusters. For each train sample, along the value of power generation, another output value is assigned, which is a one-hot encoded clustering assignment vector. Meaning, for each train sample, a vector of length N_C is created with value one in the element corresponding to the assigned cluster and zero values in the rest of the vector. These vectors are used as additional output variables of the proposed model and are responsible for training the Supervised Clustering component.

Since the clustering process is transformed to a categorical classification task, we choose the categorical cross-entropy as the minimization objective function and a "softmax" activation layer for estimating the clustering participation weights. Since a "softmax" activation function is used for estimating the clustering participation weights, these weights can be also interpreted as the probabilities of a train sample to belong to each cluster.

Categorical cross-entropy is a loss function that is used in multi-class classification tasks where a train sample can only belong to one out of many possible categories. The Categorical cross-entropy is defined as:

$$L_{CE}^j = \frac{1}{N_C}\sum_{r=1}^{N_C} t_r^j log(c_r^j) : \forall j \in N \tag{9}$$

where t_r^j is the r^{th} element in the j^{th} cluster vector and c_r^j is the clustering participation weight of the j^{th} train sample in the r^{th} cluster.

B. Final Parametric Probabilistic Forecasting

Once the clustering participation weights from the supervised clustering component are computed, the values of the Kumaraswamy distribution for the final estimation are calculated as shown in (10).

$$a_f^j = \sum_{i=1}^{N_C} c_i^j a_i^j \quad , \quad b_f^j = \sum_{i=1}^{N_C} c_i^j b_i^j \quad : \forall j \in N \quad s.t \sum_{i=1}^{N_C} c_i^j = 1 \tag{10}$$

As described in the previous section, the clustering participation weights are calculated through a "softmax" function in order for the clustering participation weights to be expressed as the probability of an input vector to be included in each cluster. For clustering purposes, instead of the "softmax" function, the "argmax" or "hardmax" function could also be used, but not in a gradient descent manner since it is not differentiable. The "argmax" function returns the value one for the clustering participation weight with the highest probability and zero for the rest of the clusters. On the other hand, the "softmax" function, as a softer and differentiable approach to "argmax", is also able to keep the information about the other, non-maximal, participation weights. Therefore, as it is concluded by (10), the final parameters often will consider, the parameters of a sub-model from a different cluster, but with a much lower weight. Therefore, we can assume that the sub-models will be focused on the cluster of training data that are assigned to, but

also have some knowledge on the situation on the rest of the clusters. This is an acceptable and encouraged behaviour of the model, especially for the cases of train samples that are on the limits of neighbouring clusters and the cluster assignment is not straightforward.

C. Multi-objective Loss Function

Within the context of this paper, a variation of stacked ensemble model is proposed with a single training phase both for the ensemble of models and the Meta-Learner. In most stacked ensemble models, the training phase has two stages. The first stage includes the training of multiple models minimizing a single loss function every time for each model. The second stage is the training of the Meta-Learner, by freezing the sub-models' parameters and using them as input in the Meta-Learner, in order to compute the best way to synthesize the output values for the stacked ensemble model.

For a single training phase, a multi-objective loss function is constructed. The final loss function, as given in (11), includes two components: the loss objective function for the final parametric probabilistic prediction L and the loss objective function for the supervised clustering L_{CE}. In this case, the L function denotes the numerical quadrature approximation of CRPS. The two terms have equal weights of $\frac{1}{2}$.

$$J = \frac{1}{2N}\sum_{j=1}^{N}[L(F(a_f^j, b_f^j), y^j) + L_{CE}^j] \tag{11}$$

In order to explain how each sub-model is trained on specific train samples, conditionally to the clustering, we examine the update functions of one of the synaptic weights used to calculate parameter a and one of the synaptic weights used to calculate parameter b of the i^{th} sub-model in (12) and (13) respectively. The update functions are based on the chain-rule of derivations.

$$\begin{aligned} w_{a,i}^{t+1} &= w_{a,i}^t - \frac{\lambda}{N}\sum_{j=1}^{N}\frac{\partial J^j}{\partial w_{a,i}^t} \\ &= w_{a,i}^t - \frac{\lambda}{N}\sum_{j=1}^{N}\frac{\partial J^j}{\partial L^j}\frac{\partial L^j}{\partial F^j}\frac{\partial F^j}{\partial a_f^{j,t}}\frac{\partial a_f^{j,t}}{\partial a_i^{j,t}}\frac{\partial a_i^{j,t}}{\partial w_{a,i}^{j,t}} \\ &= w_{a,i}^t - \frac{\lambda}{N}\sum_{j=1}^{N}\frac{\partial J^j}{\partial L^j}\frac{\partial L^j}{\partial F^j}\frac{\partial F^j}{\partial a_f^{j,t}}\frac{\partial a_i^{j,t}}{\partial w_{a,i}^{j,t}}c_i^{j,t} \\ &\forall i \in N_C, \forall t \in N_{epochs} - 1 \end{aligned} \tag{12}$$

$$\begin{aligned} w_{b,i}^{t+1} &= w_{b,i}^t - \frac{\lambda}{N}\sum_{j=1}^{N}\frac{\partial J^j}{\partial w_{b,i}^t} \\ &= w_{b,i}^t - \frac{\lambda}{N}\sum_{j=1}^{N}\frac{\partial J^j}{\partial L^j}\frac{\partial L^j}{\partial F^j}\frac{\partial F^j}{\partial b_f^{j,t}}\frac{\partial b_f^{j,t}}{\partial b_i^{j,t}}\frac{\partial b_i^{j,t}}{\partial w_{b,i}^{j,t}} \\ &= w_{b,i}^t - \frac{\lambda}{N}\sum_{j=1}^{N}\frac{\partial J^j}{\partial L^j}\frac{\partial L^j}{\partial F^j}\frac{\partial F^j}{\partial b_f^{j,t}}\frac{\partial b_i^{j,t}}{\partial w_{b,i}^{j,t}}c_i^{j,t} \\ &\forall i \in N_C, \forall t \in N_{epochs} - 1 \end{aligned} \tag{13}$$

The update functions of the synaptic weights in the rest of the layers in the sub-models are expressed in a similar way to (12) and (13). As we can see, the partial derivatives of the final parameters in respect to the parameters of the sub-models, based on (10), are equal to the clustering participation weight (red terms). Therefore, the synaptic weights are updated based on their contribution to the prediction error, but weighted based on the clustering participation weights. This leads to selected optimization of the synaptic weights of a sub-model also based on the degree of their participation in each cluster.

Furthermore, the update functions of the synaptic weights in the Supervised Clustering component, if we use again the chain-rule of derivations, are dependent both on the partial derivatives of functions L and L_{CE}. This will lead for the clustering process, not only be optimized on the clustering objective function, but also on the regression objective function. This is the reason why the clustering process within the proposed model is considered supervised and not exclusively unsupervised.

V. Case Study

In this paper, we validate the proposed approach for both wind power forecasting and solar power forecasting in three cases. Wind power generation data for Case 1 are provided by the data-set of the GEFCom 2014 [36]. Case 2 refers to solar power generation data also provided by the data-set of the Global Energy Forecasting Competition 2014. In Case 3 data from solar power plants in France are used, provided by the Smart4RES project under the Horizon 2020 Framework Program [37].

A. Dataset Description

The datasets used for validation in Cases 1 and 2 are open and provided by GEFCom 2014. These datasets provide numerical weather predictions as listed in Table I. The data are collected in 2012 and 2013 and contain hourly wind power measurements from ten different wind farms and three solar parks. For the validation process we randomly select four wind farms and all three solar parks. The data for the training process cover the period from $01/01/2012$ to $30/06/2013$ including 13104 training samples. The data for the validation process cover the period from $01/07/2013$ to $01/12/2013$ including 2304 validation samples.

The datasets used in Case 3 are provided by the Smart4RES project and include the hourly solar power generation measurements from two solar parks located in France. As input data, for these cases, the numerical weather predictions of the temperature, cloud coverage and solar radiation from the European Centre for Medium-Range Weather Forecasts model were used. Additionally, date and time information was also used as input feature (hour and month). The data for the training process cover the period from $16/12/2018$ to $31/03/2020$ including 13104 training samples. The data for the validation process cover the period from $02/04/2020$ to $30/09/2020$ including 4344 validation samples. In all cases, the training samples are split into two subsets containing the 80% and 20% respectively. The first subset is used for training and the second for testing purposes within the training process. The information of all cases is summarized in Table I.

TABLE I: Case Studies Information

Cases	Input Features	Forecasting Horizon (hours)	Source	Training Period	Validation Period
1	Wind speed (10m,100m), Wind direction (10m,100m)	24	GEFCom 2014	01/01/2012 - 30/06/2013	01/07/2013 - 01/12/2013
2	Relative humidity, Total cloud cover, Temperature, Downward solar radiation, Downward thermal radiation, Hour, Month	24	GEFCom 2014	01/01/2012 - 30/06/2013	01/07/2013 - 01/12/2013
3	Temperature, Total cloud cover, Downward solar radiation, Hour, Month	24	Smart4RES	16/12/2018 - 31/03/2020	02/04/2020 - 30/09/2020

B. Assessment Metrics and Benchmarks

Within this paper, reliability diagrams and prediction intervals' width are used to assess the reliability and sharpness of the parametric probabilistic forecasts in all cases. Furthermore, in order to asses the comprehensive quality of the predicted probability distributions, the CRPS is used, as suggested in [38]. All of the assessment metrics are averaged over the entire validation dataset of each case. In order to present an objective evaluation of the proposed model's performance several state-of-the-art models are used as benchmarks. Based on the literature, the BELM, GP, MDN are selected. The MDN model is tested in three popular variations, one with Gaussian kernels (MDN-Gaussian), one with Logit-Normal kernels (MDN-LNormal) and the IDMDN, as proposed in [20] using Beta kernels. In the case of GP, as benchmark, the recently proposed Sparse Variational Gaussian process (SVGP) as studied in [24], is used. Additionally, we include two popular models, the KDE and QRGBM as benchmarks, since they are proved to be effective in the GEFCom 2014. The QRGBM is a tree based ensemble model, that minimizes the quantile loss by iteratively adding new tree models in the ensemble. In the case of KDE, we locate the 100 nearest neighbors of each test sample and use their corresponding wind power values to estimate the predictive probability distribution. In addition, the climatology model is adopted as a naive benchmark model, which estimates the predictive probability density using all training data. Furthermore, it is necessary to prove the superiority of the CRPS used as a loss function, in comparison to the MLE, through the evaluation process. Therefore, in addition to the rest of the benchmark models, within the evaluation process, the proposed architecture was trained using the MLE and CRPS in two different models. For simplicity, these two variations will be referred to as "Proposed-MLE" and "Proposed" respectively.

VI. Results and Discussion

In this section, results of the overall performance of the model as well as the performance of the individual components are presented. Firstly we evaluate the goodness-of-fit of the Kumaraswamy distribution in solar and wind power forecasting by comparing its performance to the most common probability distributions used in RES generation probabilistic forecasting. Additionally, we evaluate the performance of the CRPS metric used as an objective function in comparison to the MLE. Furthermore, we evaluate and discuss the added value of the supervised clustering process inside the Meta-Learner component and finally the performance metrics for the proposed methodology.

TABLE II: Goodness-of-fit metrics for solar power forecasting

Metric	Kumaraswamy	Normal	Weibulll	Beta	Logit-Normal
Akaike Information criterion	-1309.33	-913.63	-91.04	-108.29	-56.33
Cramér–von Mises criterion	0.0018	0.035	0.0721	0.1753	0.2661
Kolmogorov–Smirnov test	0.6806	0.7024	0.6946	0.7023	0.8177

TABLE III: Goodness-of-fit metrics for wind power forecasting

Metric	Kumaraswamy	Normal	Weibulll	Beta	Logit-Normal
Akaike Information criterion	-223.65	-138.61	-208.29	-223.84	-65.51
Cramér–von Mises criterion	0.003	0.0199	0.0037	0.0057	0.094
Kolmogorov–Smirnov test	0.2074	0.3946	0.2124	0.2136	0.4236

A. Evaluation of Kumaraswamy Distribution

In order to support our assumption that the Kumaraswamy distribution can be used for both wind and solar power probabilistic forecasting, a goodness-of-fit test is presented. The test is based on how well the Kumaraswamy distribution can describe a probability histogram created by historical data, for a given prediction. The histogram is created using the k-nearest algorithm based on the input features(e.g weather variables, date related information) and by grouping the corresponding power measurements in a histogram. More details about the used data are provided in Table I. Along the Kumaraswamy distribution the Beta, Normal, Weibulll and Logit-normal distributions are tested. As goodness-of-fit metrics, the Akaike information criterion [39], the Cramér–von Mises criterion [40] and the Kolmogorov–Smirnov test [41] are used. In both cases of solar and wind power, the Kumaraswamy distribution achieves a better score of goodness-of-fit in comparison to the rest of the distributions, supporting the hypothesis of using that distribution for both solar and wind power probabilistic forecasting, as shown in tables II and III.

B. Evaluation of CRPS Objective Function

This section provides a comprehensive evaluation of the CRPS as an alternative objective function compared to the MLE, which is one of the most commonly used objective functions. For this evaluation, the proposed model is trained using both the CRPS and MLE metrics. For each metric and test case, the parametric probabilistic forecasts in the form of Kumaraswamy parameters are estimated. The goodness-of-fit

TABLE IV: Goodness-of-fit metrics for CRPS and MLE

Metric	CRPS	MLE
Cramér–von Mises Criterion	0.001560	0.001586
Kolmogorov-Smirnov Test	0.665824	0.671399

TABLE V: Goodness-of-fit metrics for CRPS and MLE

Metric	CRPS	MLE
Cramér–von Mises Criterion	0.000681	0.000695
Kolmogorov-Smirnov Test	0.536501	0.544291

TABLE VI: Goodness-of-fit metrics for CRPS and MLE

Metric	CRPS	MLE
Cramér–von Mises Criterion	0.001322	0.001356
Kolmogorov-Smirnov Test	0.650095	0.657783

metrics, namely the Cramer–von Mises and the Kolmogorov-Smirnov tests, are then computed and averaged. The results for three test cases are presented in Tables IV,V and VI respectively.

The results of this study demonstrate that the CRPS metric, employed as an objective function, provides superior performance in comparison to the MLE metric. The findings suggest that utilizing the CRPS in parametric probabilistic forecasting of RES generation results in distributions that offer better representation of the target variable as compared to the distributions generated by the proposed model utilizing the MLE as an objective function.

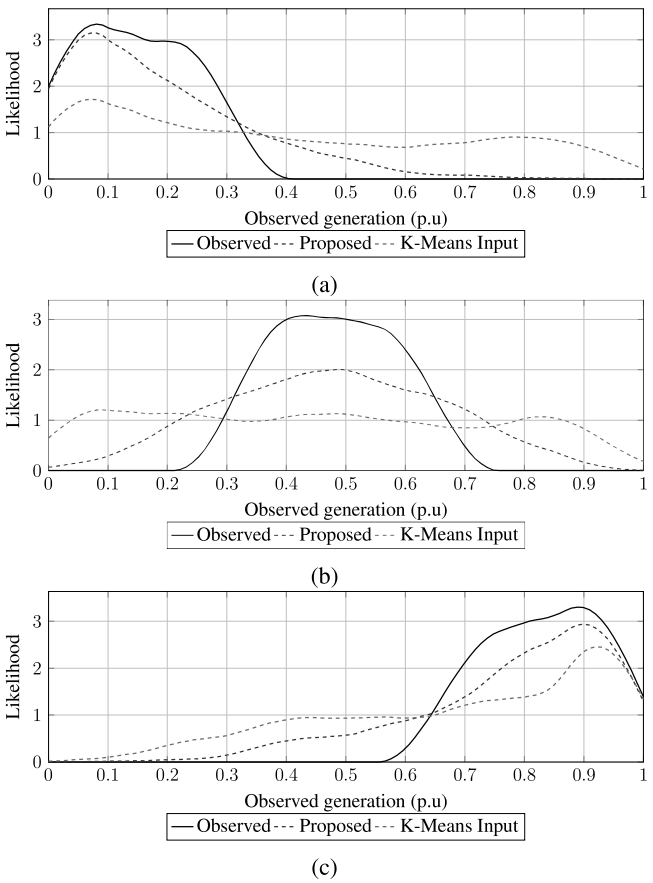


Fig. 4: Clusters' distributions of observations using the proposed clustering process, K-means clustering on the input data and the posterior grouping of observed values: (a) Cluster 1, (b) Cluster 2, (c) Cluster 3.

C. Clustering Evaluation

Within the proposed methodology, the clustering process is added in order to assist in the training of the sub-models. The main idea is to perform clustering on the output variables. This means creating clusters of input features with similar output values. This is needed because in RES generation forecasting, the input data are stochastic, i.e weather predictions are stochastic and noisy, leading to similar input values with very different output. This can mislead the training of the forecasting models, that try to capture all possible outcomes reducing their accuracy. Moreover, by applying this method, the sub-models are forced to limit their output in a specific sub-domain, minimizing in this way, the forecasting error.

To apply this clustering method, the model has to estimate "a priori" the cluster of the estimated output. In fact, if not designed properly, the introduction of clustering information to the Meta-Learner can introduce noise and reduce the overall performance of the model. For further evaluation, two variations of the presented approach have been examined: the proposed model without any clustering information for the Meta-Learner and the proposed model with offline clustering information, through K-means clustering, provided to the Meta-Learner. The supervised clustering method achieved a $2-3\%$ improvement, on average, over all the metrics and cases, in comparison to the first variation and a $5-6\%$ improvement, on average, in comparison to the second variation. Therefore, the overall performance of the proposed model, is critically connected to the performance of the supervised clustering component to estimate correctly the sub-domain in which the actual generation will be observed. In Figure 4, the distributions of observed generation values using the k-means clustering algorithm on the input data and the supervised clustering process after the training of the model, on a specific case study are displayed. These distributions are compared to the distributions of observations with a posterior knowledge of observation. As it can be observed, the supervised clustering process, achieves a superior performance over the unsupervised clustering on the input data, in regard to the grouping of train data based on similar values of generation. In general, the supervised clustering process achieves $70-80\%$ accuracy on the classification task of the output generation, proving that clustering based on similar target values instead of input features can be achieved.

D. Continuous ranked probability score

The CRPS values for Cases 1, 2 and 3 are presented in Table VII. As expected, all the benchmark models and the proposed model outperform the climatology model, due to its

TABLE VII: CRPS values in Cases 1,2 and 3 (normalized by the installed capacity)

Case 1										
Wind Farm	**Climatology**	**BELM**	**KDE**	**QRGBM**	**SVGP**	**MDN-Gaussian**	**MDN-LNormal**	**IDMDN**	**Proposed-MLE**	**Proposed**
1	0.1731	0.1091	0.0953	0.0979	0.0953	0.0945	0.0941	0.0937	0.0948	0.0931
2	0.1815	0.0947	0.0813	0.0824	0.0835	0.0818	0.0821	0.0816	0.0821	0.0810
3	0.1959	0.1013	0.0884	0.0887	0.0843	0.0870	0.0894	0.0833	0.0847	0.0802
4	0.1613	0.0827	0.0732	0.0744	0.0754	0.0784	0.0735	0.0771	0.0753	0.0723
Case 2										
Solar Park	**Climatology**	**BELM**	**KDE**	**QRGBM**	**SVGP**	**MDN-Gaussian**	**MDN-LNormal**	**IDMDN**	**Proposed-MLE**	**Proposed**
1	0.1482	0.0724	0.0735	0.0512	0.0536	0.0572	0.0498	0.0557	0.0527	0.0507
2	0.1539	0.0753	0.074	0.0579	0.0548	0.0559	0.0528	0.0537	0.0535	0.0523
3	0.1547	0.0734	0.0752	0.0642	0.0535	0.0534	0.0522	0.0530	0.0528	0.0517
Case 3										
Solar Park	**Climatology**	**BELM**	**KDE**	**QRGBM**	**SVGP**	**MDN-Gaussian**	**MDN-LNormal**	**IDMDN**	**Proposed-MLE**	**Proposed**
1	0.1852	0.0784	0.0752	0.0675	0.0701	0.0836	0.0731	0.0682	0.0661	0.0666
2	0.1726	0.0841	0.0892	0.0781	0.0765	0.0865	0.0766	0.0774	0.0799	0.0758
3	0.1104	0.0641	0.0594	0.0585	0.0533	0.0604	0.0571	0.0556	0.0548	0.0524

simplicity. Additionally, all benchmark models perform well in both solar and wind power generation forecasting in the case of CRPS metric, therefore, it can be concluded that the different benchmark models can provide a very accurate and detailed comparison with the performance of the proposed model.

The CRPS performance of the benchmark models is relatively comparable to the performance of the proposed model, indicating that the distribution assumption of the proposed model has not limited its performance. On the contrary, the proposed model outperforms all the benchmark models, in most cases, as it has the lowest CRPS values in the evaluation process. This indicates that the use of a parametric model that includes the CRPS in its minimization objective function, instead of MLE, can actually achieve lower values of CRPS. Furthermore, from the results it can be seen that the proposed method is able to perform equally good and in most cases better in comparison to the MDN variations and SVGP. This is a great success of the proposed method as it can achieve state-of-the-art performance while solving some of the drawbacks introduced from the rest of the benchmark models as discussed in the first section of this paper.

E. Reliability and sharpness of prediction intervals

For each case, the reliability diagram and prediction intervals' width for a single sub-case are displayed. In Figures 5, 6 and 7 the reliability diagrams and prediction intervals' widths for Cases 1, 2 and 3 are displayed respectively. In Case 1 it can be seen that the SVGP, the MDN variations and the proposed model are proved to be greatly reliable and close to the ideal reliability. This is a remarkable achievement of the proposed model, since its goal is to estimate an entire distribution based on minimizing the CRPS. In Cases 2 and 3, which are solar power forecasting tasks, it is observed that the performance of the benchmark models varies, with the proposed method being the most reliable and is able to achieve close to ideal reliability with the narrowest prediction intervals, in comparison to the benchmark models.

By examining the performance of the proposed model, in all three metrics, we can conclude that the use of ensemble branches or sub-models, within the same model, where each branch is trained based on the level of generated power, the proposed model can achieve close to ideal reliability on parametric estimation for the output. The Meta-Learner component by choosing the correct sub-model, conditionally to the input data, is able to predict the shape of the probability distribution of the outcome and respect both the reliability of the estimation and the boundaries of the power generation, while minimizing the width of the distribution and the uncertainty of the prediction.

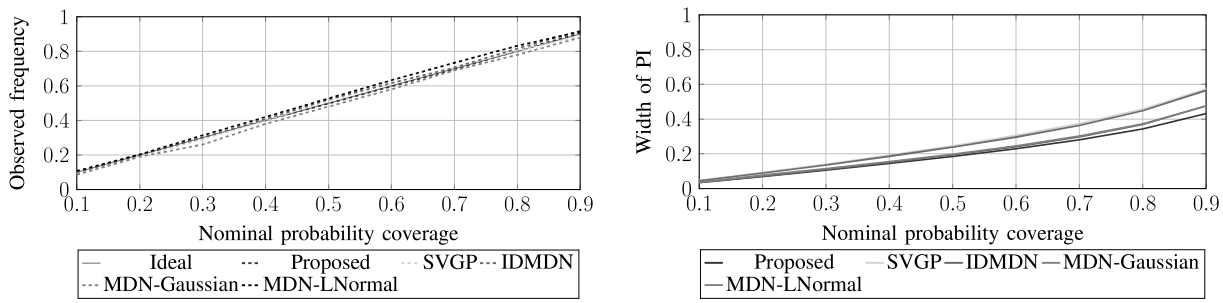
VII. Conclusion

In conclusion, this paper proposes a novel probabilistic method, based on artificial neural networks. The method combines the processes of clustering ensemble forecasting and probability density estimation in a single multi-component architecture. Every process is held by an individual component, which minimizes its own objective function in a single training phase. These components are interconnected, creating a multi-objective function to be minimized. The main idea of the proposed methodology is to create an ensemble of probabilistic sub-models and through a clustering process decide in the final component, which sub-models will participate in the final estimation, conditionally to the input and the cluster to which they belong to. The probabilistic performance of the proposed model is evaluated and compared to multiple commonly used and state-of-the-art forecasting models in three test cases.

The proposed methodology is proven to outperform all the benchmark models, for probabilistic wind and solar power generation forecasting for day-ahead horizons with hourly time resolution. The input data used are in general numerical weather predictions, which are commonly available to RES plant operation engineers. These weather predictions include wind speed and direction for wind power forecasting and temperature, cloud coverage and solar radiation for solar power forecasting. Additionally, the date and time information are provided as input data. The simplicity of the input data, makes the proposed model easily applicable in most cases of day-ahead RES generation forecasting.

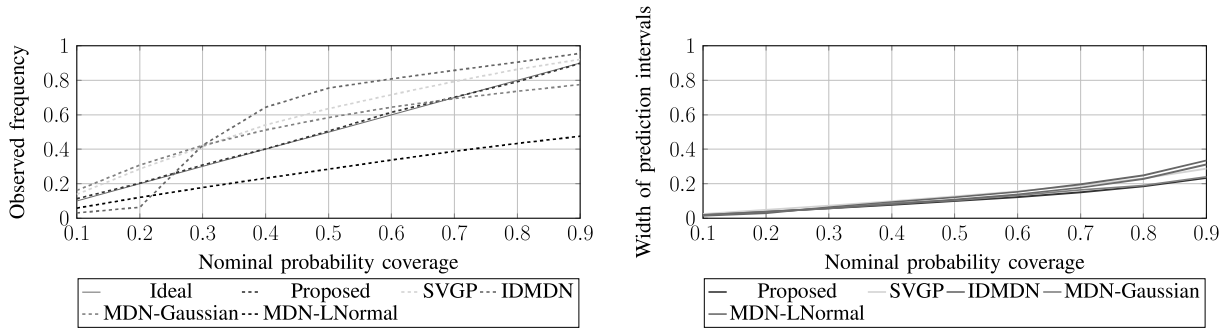
VIII. Acknowledgement

This paper is supported by European Union's Horizon 2020 research and innovation program under grant agreement No 864337, project Smart4RES.



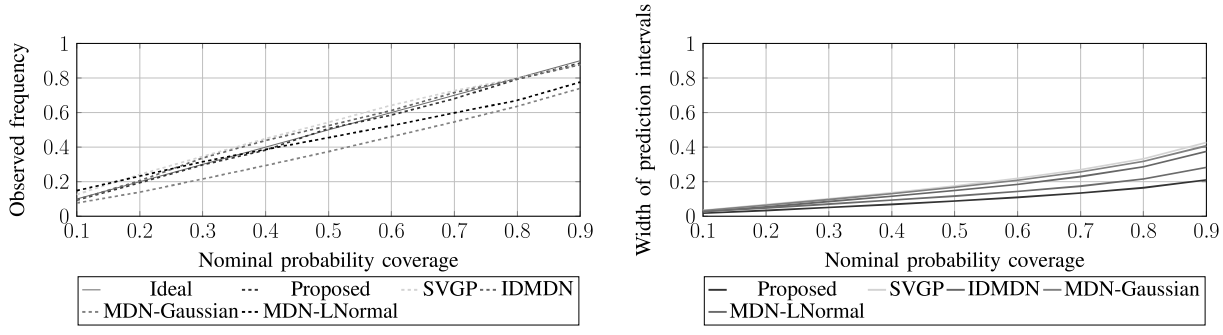
(a) (b)

Fig. 5: Case 1 - Wind farm 3: (a) Reliability diagram of forecasts (b) Width of prediction intervals.



(a) (b)

Fig. 6: Case 2 - Solar park 1: (a) Reliability diagram of forecasts (b) Width of prediction intervals.



(a) (b)

Fig. 7: Case 3 - Solar park 1: (a) Reliability diagram of forecasts (b) Width of prediction intervals.

REFERENCES

[1] P. Samadi, V. W. S. Wong, and R. Schober, "Load scheduling and power trading in systems with high penetration of renewable energy resources," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 1802–1812, 2016.

[2] M. S. Alam, F. S. Al-Ismail, M. A. Abido, and A. Salem, "High-level penetration of renewable energy with grid: Challenges and opportunities," 2020.

[3] O. Erdinc, N. G. Paterakis, and J. P. Catalão, "Overview of insular power systems under increasing penetration of renewable energy sources: Opportunities and challenges," *Renewable and Sustainable Energy Reviews*, vol. 52, pp. 333 – 346, 2015.

[4] P. Pinson, C. Chevallier, and G. N. Kariniotakis, "Trading wind generation from short-term probabilistic forecasts of wind power," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1148–1156, 2007.

[5] J. Yan, Y. Liu, S. Han, Y. Wang, and S. Feng, "Reviews on uncertainty analysis of wind power forecasting," *Renewable and Sustainable Energy Reviews*, vol. 52, pp. 1322 – 1330, 2015.

[6] A. Zakaria, F. B. Ismail, M. H. Lipu, and M. Hannan, "Uncertainty models for stochastic optimization in renewable energy applications," *Renewable Energy*, vol. 145, pp. 1543 – 1571, 2020.

[7] J. Morales, A. Conejo, H. Madsen, P. Pinson, and M. Zugno, *Integrating Renewables in Electricity Markets - Operational Problems*, 01 2014.

[8] J. B. Bremnes, "A comparison of a few statistical models for making quantile wind power forecasts," *Wind Energy*, vol. 9, pp. 3 – 11, 04 2006.

[9] J. M. Morales, A. J. Conejo, and J. Perez-Ruiz, "Economic valuation of reserves in power systems with high penetration of wind power," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 900–910, 2009.

[10] Z. Wang, C. Shen, L. Feng, X. Wu, C.-C. Liu, and F. Gao, "Chance-constrained economic dispatch with non-gaussian correlated wind power uncertainty," *IEEE Transactions on Power Systems*, vol. PP, pp. 1–1, 02 2017.

[11] C. Wan, J. Lin, J. Wang, Y. Song, and Z. Y. Dong, "Direct quantile regression for nonparametric probabilistic forecasting of wind power generation," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2767–2778, 2017.

[12] Y.-K. Wu, P.-E. Su, T.-Y. Wu, J.-S. Hong, and M. Y. Hassan, "Probabilistic wind-power forecasting using weather ensemble models," *IEEE*

Transactions on Industry Applications, vol. 54, no. 6, pp. 5609–5620, 2018.

[13] A. Kavousi-Fard, A. Khosravi, and S. Nahavandi, "A new fuzzy-based combined prediction interval for wind power forecasting," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 18–26, 2016.

[14] G. Sideratos and N. D. Hatziargyriou, "Probabilistic wind power forecasting using radial basis function neural networks," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 1788–1796, 2012.

[15] M. Landry, T. P. Erlinger, D. Patschke, and C. Varrichio, "Probabilistic gradient boosting machines for gefcom2014 wind forecasting," *International Journal of Forecasting*, vol. 32, no. 3, pp. 1061–1066, 2016.

[16] W. Dong, H. Sun, J. Tan, Z. Li, J. Zhang, and H. Yang, "Regional wind power probabilistic forecasting based on an improved kernel density estimation, regular vine copulas, and ensemble learning," *Energy*, vol. 238, p. 122045, 2022.

[17] Y. Zhang and J. Wang, "K-nearest neighbors and a kernel density estimator for gefcom2014 probabilistic wind power forecasting," *International Journal of Forecasting*, vol. 32, no. 3, pp. 1074 – 1080, 2016.

[18] R. J. Bessa, V. Miranda, A. Botterud, J. Wang, and E. M. Constantinescu, "Time adaptive conditional kernel density estimation for wind power forecasting," *IEEE Transactions on Sustainable Energy*, vol. 3, no. 4, pp. 660–669, 2012.

[19] C. M. Bishop, "Mixture density networks," 1994.

[20] H. Zhang, Y. Liu, J. Yan, S. Han, L. Li, and Q. Long, "Improved deep mixture density network for regional wind power probabilistic forecasting," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2549–2560, 2020.

[21] H. Wen, P. Pinson, J. Ma, J. Gu, and Z. Jin, "Continuous and distribution-free probabilistic wind power forecasting: A conditional normalizing flow approach," *IEEE Transactions on Sustainable Energy*, vol. 13, no. 4, pp. 2250–2263, 2022.

[22] J. Yan, K. Li, E.-W. Bai, J. Deng, and A. M. Foley, "Hybrid probabilistic wind power forecasting using temporally local gaussian process," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 1, pp. 87–95, 2016.

[23] P. Kou, F. Gao, and X. Guan, "Sparse online warped gaussian process for wind power probabilistic forecasting," *Applied Energy*, vol. 108, pp. 410–428, 2013.

[24] H. Wen, J. Ma, J. Gu, L. Yuan, and Z. Jin, "Sparse variational gaussian process based day-ahead probabilistic wind power forecasting," *IEEE Transactions on Sustainable Energy*, vol. 13, no. 2, pp. 957–970, 2022.

[25] F. Golestaneh, P. Pinson, and H. B. Gooi, "Very short-term nonparametric probabilistic forecasting of renewable energy generation— with application to solar energy," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3850–3863, 2016.

[26] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1033–1044, 2014.

[27] T. Konstantinou, N. Savvopoulos, and N. Hatziargyriou, "Post-processing numerical weather prediction for probabilistic wind forecasting," in *2020 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, 2020, pp. 1–6.

[28] H. Li, J. Deng, P. Feng, C. Pu, D. D. Arachchige, and Q. Cheng, "Short-term nacelle orientation forecasting using bilinear transformation and iceemdan framework," in *Frontiers in Energy Research*, 2021.

[29] S. Ghimire, R. C. Deo, D. Casillas-Pérez, and S. Salcedo-Sanz, "Improved complete ensemble empirical mode decomposition with adaptive noise deep residual model for short-term multi-step solar radiation prediction," *Renewable Energy*, vol. 190, pp. 408–424, 2022.

[30] W. Wu and M. Peng, "A data mining approach combining k -means clustering with bagging neural network for short-term wind power forecasting," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 979–986, 2017.

[31] "Deep belief network based k-means cluster approach for short-term wind power forecasting," *Energy*, vol. 165, pp. 840–852, 2018.

[32] L. Dong, L. Wang, S. F. Khahro, S. Gao, and X. Liao, "Wind power day-ahead prediction with cluster analysis of nwp," *Renewable and Sustainable Energy Reviews*, vol. 60, pp. 1206–1212, 2016.

[33] A. Kusiak and W. Li, "Short-term prediction of wind power with a clustering approach," *Renewable Energy*, vol. 35, no. 10, pp. 2362–2369, 2010.

[34] M. Jones, "Kumaraswamy's distribution: A beta-type distribution with some tractability advantages," *Statistical Methodology*, vol. 6, no. 1, pp. 70–81, 2009.

[35] M. Gebetsberger, J. W. Messner, G. J. Mayr, and A. Zeileis, "Estimation methods for nonhomogeneous regression models: Minimum continuous ranked probability score versus maximum likelihood," *Monthly Weather Review*, vol. 146, no. 12, pp. 4323 – 4338, 2018.

[36] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond," *International Journal of Forecasting*, vol. 32, no. 3, pp. 896 – 913, 2016.

[37] https://www.smart4res.eu/.

[38] J. Messner, P. Pinson, J. Browell, M. Bjerregård, and I. Schicker, "Evaluation of wind power forecasts—an up-to-date view," *Wind Energy*, vol. 23, no. 6, pp. 1461–1481, Jan. 2020.

[39] K. Burnham, "Understanding aic and bic in model selection," *Sociological Methods & Research - SOCIOL METHOD RES*, vol. 33, p. 93, 11 2004.

[40] H. Cramér, "On the composition of elementary errors: First paper: Mathematical deductions," *Scandinavian Actuarial Journal*, vol. 1928, no. 1, pp. 13–74, 1928.

[41] "Kolmogorov–smirnov test," pp. 283–287, 2008.