An Interval-Valued Neural Network Approach for Prediction Uncertainty Quantificationin Short-term Wind Speed Prediction

Ronay Ak, Valeria Vitelli, and Enrico Zio, *Senior Member*, *IEEE*

***Abstract*—We consider the task of performing prediction with neural networks on the basis of uncertain input data expressed in the form of intervals. We aim at quantifying the uncertainty in the prediction arising from both the input data and the prediction model. A multi-layer perceptron neural network (NN) is trained to map interval-valued input data into interval outputs, representing the prediction intervals (PIs) of the real target values. The NN training is performed by non-dominated sorting genetic algorithm–II (NSGA-II), so that the PIs are optimized both in terms of accuracy (coverage probability) and dimension (width). Demonstration of the proposed method is given on two case studies: (i) a synthetic case study, in which the data have been generated with a 5-min time frequency from an Auto-Regressive Moving Average (ARMA) model with either Gaussian or Chi-squared innovation distribution; (ii) a real case study, in which experimental data consist in wind speed measurements with a time-step of 1-hour. Comparisons are given with a crisp (single-valued) approach. The results show that the crisp approach is less reliable than the interval-valued input approach in terms of capturing the variability in input.**

***Index Terms*—Interval-valued neural networks, multi-objective genetic-algorithm, prediction intervals, short-term wind speed forecasting, uncertainty.**

I. Introduction

PREDICTION plays a crucial role in every decision-making process, and for this reason it should take into account any source of uncertainty that may affect its outcome. Prediction uncertainty can arise due to measurement errors,lack of knowledge in input data, and model approximation errors (e.g. due to imperfections in the model formulation) [1]-[3].For practical purposes, uncertainties can be classified in two distinct types [3]: epistemic and aleatory. The former derives from imprecise model representation of the system behavior, in terms of uncertainty in both the hypotheses assumed (structural uncertainty) and the values of the model parameters (parameter uncertainty) [4]. The latter describes the inherent variability of the observed physical phenomenon, and it is therefore also named stochastic uncertainty, irreducible uncertainty, or inherent uncertainty [5].

Uncertainty quantification is the process of representing the uncertainty in thesystem inputs and parameters, propagating it through the model, and then revealing the resulting uncertainty in themodel outcomes [2].

In the literature, methods such as evidence theory [5], probability modeling [6],Neural Networks-based prediction intervals estimation [7]-[11],conformal prediction [12], [13], interval analysis [14]-[16], fuzzy set theory [17], and in particular type-2 fuzzy sets and systems [18]-[21], as well as extensions of fuzzy mathematical morphology [22], [23], Monte Carlo simulation [24], and Latin hypercube sampling [25] have been used to efficiently represent, aggregate, and propagate different types of uncertainty through computational models. Interval analysis is a powerful technique for bounding solutions under uncertainty. The uncertain model parameters are described by upper and lower bounds, and the corresponding bounds in the model output are computed using interval functions and interval arithmetic [26]. These bounds contain the true target value with a certain confidence level. The interval-valued representation can also be used to reflect the variability in the inputs (e.g. extreme wind speeds in a given area, minimum and maximum of daily temperature, etc.), or their associated uncertainty (e.g. strongly skewed wind speed distributions, etc.), i.e. to express the uncertain information associated to the input parameters[14]-[16], [27].

In this paper, we present an interval-valued time series prediction modelingframework based on a data-driven learning approach, more specifically a multi-layer perceptron neural network (NN). Demonstration of the proposed method is given on two case studies: (i) a synthetic case study, with 5-minutes simulated data; (ii) a real case study,involving hourly wind speed measurements. In both cases, short-term prediction (1-hour and day-ahead, respectively) is performed taking into account both the uncertainty in the model structure, and the variability (within-hour and within-day, respectively) in the inputs.

The wind speed prediction case study has been chosen because of its relevance for wind power production. Among the various renewable energy candidates, wind energy has

R. Ak and E. Zio are with Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricité de France, Châtenay-Malabry 92290 and Gif-Sur-Yvette 91192, France (e-mail: ronay.ak@supelec.fr; enrico.zio@ecp.fr).

V. Vitelli is with the Department of Biostatistics, University of Oslo, Oslo, Norway (e-mail:valeria.vitelli@medisin.uio.no).

E. Zio is with the Department of Energy, Politecnico di Milano, Milan 20133, Italy (e-mail: enrico.zio@ecp.fr).

received fast growing attention throughout the world, and the utilization of wind power has increased dramatically over the past decade: the worldwide wind capacity has reached 296 GW by the end of June 2013, out of which 13980 MW have been added in that first half of 2013 [28]. This increasing integration of wind energy into power grid leads to additional uncertainty in the system due to the stochastic characteristics of wind itself. Wind power variations in short-term time scales have significant effects on power system operations such as regulation, load following, balancing, unit commitment and scheduling[8], [29], [30]. Thus, accurate prediction of wind speed and its uncertainty is critical for the safe, reliable and economic operation of power systems [29], [30]. In other words, optimal integration of wind power into the grid requires highly accurate predictions with a reliable assessment of the uncertainties associated to the system. To this aim, Prediction Intervals (PIs) are a simple way to communicate a measure of the uncertainty in the predictions. PIs are preferable results of the prediction, rather than point estimates, because they provide information on the confidence in the prediction [7]-[11], taking into account the underlying uncertainties.

In the present work, an interval representation has been given to the hourly and daily inputs by using two different approaches (see Section IV), which quantify in two different ways the within-hour and within-day variability. The network maps interval-valued input data into an interval output, providing the estimated prediction intervals (PIs) for the real target. PIs are comprised of lower and upper bounds within which the actual target is expected to lie with a predetermined probability[7]-[11]. The NN prediction model is trained by a multi-objective genetic algorithm (MOGA) (the powerful non-dominated sorting genetic algorithm-II, NSGA-II), so that the PIs are optimal both in terms of accuracy (coverage probability) and dimension (width).

The prediction interval coverage probability (PICP) represents the probability that the set of estimated PI values will contain a certain percentage of the true output values. Prediction interval width (PIW) simply measures the extension of the interval as the difference of the estimated upper and lower bound values. The network uses interval-valued data but its weights and biasesare crisp (i.e. single-valued). The NSGA-II training procedure generates Pareto-optimal solution sets, which include non-dominated solutions for the two objectives (PICP and PIW).

The originality of the work appears in two aspects: (i) while the existing papers on short-term wind speed/power prediction use single-valued data as inputs, obtained as a within-hour [11], [29] or within-day average [31],[32], we give an interval representation to hourly/daily inputs by using two approaches (see Section IV), which properly account (in two different ways) for the within-hour/day variability; (ii) we handle the PIs problem in a multi-objective framework[11], [33], whereas the existing relevant methods for wind speed/power prediction [8]consider only one objective for optimization.Also, the proposed approach integrates the estimation of the prediction intervals in its learning procedure while several methods construct PIs in two steps (first doing point prediction and then constructing PIs). In short, with the present work, we are able to account for both aleatoryuncertainty in wind speed and epistemic uncertainty due to model parameters and to demonstrate an indication of how the uncertainties in input affect the output quantities,by the interval representation of the input variables.

It is worth recalling that in [11], we have performed a comparison with single-objective genetic algorithm (SOGA) and single-objective simulated annealing (SOSA) methods. SOSA has been proposed in support of the LUBE method in [7]. The comparison results show that the PIs produced by NSGA-II compare well with those obtained by LUBE and are satisfactory in both objectives of high coverage and small width. In [33], we have implemented the NSGA-II to train a NN to provide the PIs of the scale deposition rate. We have performed *k*-fold cross-validation to guide the choice of the NN structure (i.e. the number of hidden neurons) with good generalization performance. We have used a hypervolume indicator metric to compare the Pareto fronts obtained in each cross-validation fold.All these analyses have been performed with single-valued inputs for both works.More precisely, in [11],single-valued historical wind speed values $W_{t-1}, W_{t-2,\dots,} W_{t-k}$have been selected as input variables for predicting W_t in output. In [33], the case study concerns the scale (deposition) rate on the metal surfaces of equipment used in offshore oil wells. The output variable is the scale rate (y), and it has been predicted using the single-valued influencing input variables: temperature (T) and pressure (P), water composition (W) and fluid velocity (V) near the metal surfaces.

The paper is organized as follows. Section II introduces the basic concepts of interval-valued NNs for PIs estimation. In Section III, basic principles of multi-objective optimization are briefly recalled and the use of NSGA-II for training a NN to estimate PIs is illustrated. Experimental results on the synthetic case study and on the real case study concerning wind speed prediction are given in Section IV. Finally, Section V concludes the paper with a critical analysis of the results and some ideas for future studies.

II. Neural networks and Prediction intervals

Neural networks (NNs) are a class of nonlinear statistical models inspired by brain architecture, capable of learning complex nonlinear relationships among variables from observed data [34]. This is done by a process of parameter tuning called "training". It is common to think of a NN model as a way of solving a nonlinear regression problem of the kind [35], [36]:

$$y = f(x; w^*) + \varepsilon(x), \quad \varepsilon(x) \sim N(0, \sigma_\varepsilon^2(x)) \quad (1)$$

where x,y are the input and output vectors of the regression, respectively, and w^* represents the vector of values of the parameters of the model function f, in general nonlinear. The term $\varepsilon(x)$ is the error associated with the regression model f, and it is assumed normally distributed with zero mean. For

simplicity of illustration, in the following we assume y one-dimensional. An estimate $\hat{w}$ of w^* can be obtained by a training procedure aimed at minimizing the quadratic error function on a training set of input/output values$D = \{(x_i, y_i), i = 1,2, \dots, n_p\}$,

$$E(w) = \sum_{i=1}^{n_p} (\hat{y}_i - y_i)^2 \quad (2)$$

where$\hat{y}_i = f(x_i; \hat{w})$ represents the output provided by the NN in correspondence of the input x_i and n_p is the total number of training samples.

A PI is a statistical estimator composed by upper and lower bounds that include a future unknown value of the target $y(x)$ with a predetermined probability, called confidence level in literature [7]-[11].

To evaluate the quality of the PIs, we take the prediction interval coverage probability (PICP) and the prediction interval width (PIW) [7], [10] as measures: the former represents the probability that the set of estimated PIs will contain the true output values $y(x)$ (to be maximized), and the latter simply measures the extension of the interval as the difference of the estimated upper bound and lower bound values (to be minimized). In general, these two measures are conflicting (i.e., wider intervals give larger coverage), but in practice it is important to have narrow PIs with high coverage probability [7].

When interval-valued data [26] are used as input, each input pattern x_i is represented as an interval $x_i = [x_i^-, x_i^+]$ where $x_i^- \leq x_i^+$ are the lower and upper bounds (real values) of the input interval, respectively. Each estimated output value $\hat{y}_i$ corresponding to the $i-th$ sample x_i is, then, described by an interval as well, $\hat{y}_i = [\hat{y}_i^-, \hat{y}_i^+]$, where $\hat{y}_i^- \leq \hat{y}_i^+$ are the estimated lower and upper bounds of the PI in output, respectively.

The mathematical formulation of the PICP and PIW measures given by [7] is modified for interval-valued input and output data:

$$PICP = \frac{1}{n_p} \sum_{i=1}^{n_p} c_i \quad (3)$$

wheren_p is the number of training samples in the considered input dataset, and

$$c_i = \begin{cases} 1 & y_i \subseteq [\hat{y}_i^-, \hat{y}_i^+] \\ \frac{diam\,(y_i \cap \hat{y}_i)}{diam\,(y_i)} & y_i \not\subset [\hat{y}_i^-, \hat{y}_i^+] \wedge y_i \cap \hat{y}_i \neq \emptyset \\ 0 & otherwise \end{cases} \quad (4)$$

where $y_i = [y_i^-, y_i^+]$,$y_i^- \leq y_i^+$ are the lower and upper bounds (true values) of the output interval, respectively, and $diam()$ indicates the width of the interval. More precisely, (4) means that if the interval-valued real target is covered by the estimated PI, i.e. if the target is a subinterval of the estimated PI, thenc_i is equal to 1. If the estimated PI does not cover the entire real target, but the intersection of the two is not empty, then c_i is equal to the ratio between $diam(y_i \cap \hat{y}_i)$ and the width of the interval y_i, and in that case c_i takes a values smaller than 1. Finally, if the estimated PI does not cover the entire real target and the intersection of the two is empty, then the coverage c_i of the $i-th$ sample is 0. Due to lack of further information, this calculation corresponds to the probabilistic assumption that the target y_i can take any value in $[y_i^-, y_i^+]$ with uniform probability, i.e. that each point in $[y_i^-, y_i^+]$ is equally likely to be a possible value of *y*.

For PIW, we consider the normalized quantity:

$$NMPIW = \frac{1}{n_p} \frac{\sum_{i=1}^{n_p} (\hat{y}_i^+ - \hat{y}_i^-)}{y_{max} - y_{min}} \quad (5)$$

where NMPIW stands for Normalized Mean PIW, and y_{min} and y_{max} represent the minimum and maximum values of the true targets (i.e., the bounds of the range in which the true values fall). Normalization of the PI width by the range of targets makes it possible to objectively compare the PIs, regardless of the techniques used for their estimation or the magnitudes of the true targets.

Note that (3) is an empirical version of PICP, which yields an estimate of PICP according to the frequentist interpretation of probability theory.Similarly, (5) yields an estimate for NMPIW.

III. Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) multi-objective optimization for Neural network training

The problem of finding PIs optimal both in terms of coverage probability and width can be formulated in a multi-objective optimization framework considering the two conflicting objectives PICP and NMPIW.

A. Multi-objective Optimization by NSGA-II

In all generality, a multi-objective optimization problem considers a number of objectives, f_m, $m = 1, 2, \dots, M$, inequality g_j, $j = 1, 2, \dots, J$ and equality h_k, $k = 1, 2, \dots, K$ constraints, and bounds on the decision variables x_i, $i = 1, 2, \dots, I$. Mathematically the problem can be written as follows [37]:

$$Minimise/Maximise\ f_m(x), \quad m = 1,2, \dots, M; \quad (6)$$

$$subject\ to \quad g_j(x) \geq 0, \quad j = 1,2, \dots, J; \quad (7)$$

$$h_k(x) = 0, \quad k = 1,2, \dots, K; \quad (8)$$

$$x_i^{(l)} \leq x_i \leq x_i^{(u)} \quad i = 1,2, \dots, I. \quad (9)$$

A solution, $x = \{x_1, x_2, \dots x_I\}$ is an I-dimensional decision variable vector in the solution space R^I, restricted by the constraints (7), (8) and by the bounds on the decision variables (9).

The search for optimality requires that the M objective functions $f_m(x)$, $m = 1, 2, \dots, M$be evaluated in correspondence of the decision variable vector x in the search space. The comparison of solutions during the search is performed in terms of the concept of dominance [37]. Precisely, in case of a minimization problem, solution x_a is

regarded to dominate solution x_b $(x_a \succ x_b)$ if the following conditions are satisfied:

$$\forall i \in \{1,2,\dots,M\}: f_i(x_a) \leq f_i(x_b) \wedge \quad (10)$$

$$\exists j \in \{1,2,\dots,M\}: f_j(x_a) < f_j(x_b) \quad (11)$$

If any of the above two conditions is violated, the solution x_a does not dominate the solution x_b, and x_b is said to be non-dominated by x_a. Eventually, the search aims at identifying a set of optimal solutions $x^* \in R^I$ whichare superior to any other solution in the search space with respect to all objective functions,and which do not dominate each other. This set of optimal solutions is called Pareto optimal set; the corresponding values of the objective functions form the so called Pareto-optimal front in the objective functions space.

In this work, we use GA for the multi-objective optimization. GA is a population based meta-heuristics inspired by the principles of genetics and natural selection [38]. It can be used for solving multi-objective optimization problems [39], [40]. Among the several options for MOGA, we adopt NSGA-II, as comparative studies show that it is very efficient [38], [40], [41].

B. Implementation of NSGA-II for training a NN for Estimating PIs

In this work, we extend the method described in [7] to a multi-objective framework for estimating output PIs from interval-valued inputs. More specifically, we use NSGA-II for finding the values of the parameters of the NN which optimize two objective functions PICP (3) and NMPIW (5) in a Pareto optimality sense (for ease of implementation, the maximization of PICP is converted to minimization by subtracting from one, i.e. the objective of the minimization is 1-PICP).

The practical implementation of NSGA-II on our specific problem involves two phases: initialization and evolution. These can be summarized as follows (for more details on the NSGA-II implementation see [33]):

1) Initialization phase:

Step 1: Split the input data into training (D_{train}) and testing (D_{test}) subsets.

Step 2: Fix the maximum number of generations and the number of chromosomes (individuals) Nc in each population; each chromosome codes a solution by G real-valued genes, where G is the total number of parameters (weights) in the NN. Set the generation number $n = 1$. Initialize the first population P_n of size Nc, by randomly generating Nc chromosomes.

Step 3: For each input vector x in the training set, compute the lower and upper bound outputs of the Nc NNs, each one with G parameters.

Step 4: Evaluate the two objectives PICP and NMPIW for the Nc NNs (one pair of values 1-PICP and NMPIW for each of the Nc chromosomes in the population P_n).

Step 5: Rank the chromosomes (vectors of G values) in the population P_n by running the fast non-dominated sorting algorithm [41] with respect to the pairs of objective values, and identify the ranked non-dominated fronts $F_1, F_2, \dots, F_k$where $F_1$is the best front, F_2 is the second best front and F_k is the least good front.

Step 6: Apply to P_n a binary tournament selection based on the crowding distance [41], for generating an intermediate population S_n of size Nc.

Step 7: Apply the crossover and mutation operators to S_n, to create the offspring population Q_n of size Nc.

Step 8: Apply Step 3 onto Q_n and obtain the lower and upper bound outputs.

Step 9: Evaluate the two objectives in correspondence of the solutions in Q_n, as in Step 4.

2) Evolution phase:

Step 10: If the maximum number of generations is reached, stop and return P_n. Select the first Pareto front F_1as the optimal solution set. Otherwise, go to Step 11.

Step 11: Combine P_n and Q_n to obtain a union population $R_n = P_n \cup Q_n$.

Step 12: Apply Steps 3-5 onto R_n and obtain a sorted union population.

Step 13: Select the Nc best solutions from the sorted union to create the next parent population P_{n+1}.

Step 14: Apply Steps 6-9 onto P_{n+1} to obtain Q_{n+1}. Set $n = n + 1$; and go to Step 10.

Finally, the best front in terms of non-dominance and diversity of the individual solutions is chosen. Once the best front is chosen, the testing step is performed on the trained NN with optimal weight values.

Note that herein the diversity corresponds to "crowding distance" [41]. Each solution $i$in the population has two attributes: nondomination ranki_{rank} and crowding distance $i_{distance}$. For a solution pair, $i$and j, we have$i \prec_n j$if $i_{rank} < j_{rank}$ or ($i_{rank} = j_{rank}$ and $i_{distance} > j_{distance}$). That is, if there are two solutions under consideration with different nondomination ranks, we prefer the one with the lower (better) rank. Otherwise, if both solutions have same ranking, i.e. belong to the same non-dominated front, we select the solution which locates in the region with the smaller number of points. Note that non-dominant solutions are found by performing the fast non-dominated sorting algorithm: the chromosomes (vectors of G values) in the population P_n are ranked by running the fast non-dominated sorting algorithm [41] with respect to the pairs of objective values, and then we identify the ranked non-dominated fronts $F_1, F_2, \dots, F_k$ where F_1 is the best front, F_2 is the second best front and F_k is the least good front. Finally, in order to obtain the optimal Pareto front to be used in practice, we take the first 50 non-dominated solutions in the first front F_1. Of course, one can select more solutions. For further explanations, we refer the readers to [41].

The total computational complexity of the proposed algorithm depends on two sub-operations: non-dominated sorting and fitness evaluation. The time complexity of non-dominated sorting is $O(MNc^2)$, where M is the number of objectives and Nc is the population size [41]. In the fitness evaluation phase, NSGA-II is used to train a NN which has n_p

input samples. Since for each individual of the population a fitness value is obtained, this process is repeated $Nc \times n_p$ times. Hence, time complexity of this phase is $O(Nc \times n_p)$. In conclusion, the computational complexity of one generation is $O\left(MNc^2 + Nc \times n_p\right)$.

IV. Experiments and Results

Two case studies have been considered: a synthetic case study, consisting of four time series datasets generated according to different input variability scenarios, and a real case study concerning time series of wind speed data. The synthetic time series datasets have been generated with a 5-min time frequency from an Auto-Regressive Moving Average (ARMA) model with either Gaussian or Chi-squared innovation distribution. For what concerns the real case study, hourly measurements of wind speed for a period of 3 years (from 2010 to 2012) related to Regina, a region of Canada, have been used [42].

The synthetic case study is aimed at considering hourly data andthe effects of within-hour variability. Hourly interval input data is obtained from the 5-min time series data by two different approaches, which we refer to as "min-max" and "mean": the former obtains hourly intervals by taking the minimum and the maximum values of the 5-min time series data within each hour; the latter, instead, obtains one-standard deviation intervals $[\overline{x_i} - s_i, \overline{x_i} + s_i]$ by computing the sample mean ($\overline{x_i}$)and standard deviation (s_i) of each 12 within-hour 5-min data sample. Single-valued (crisp) hourly input have also been obtained as a within-hour average, i.e. by taking the mean of each 12 within-hour 5-min data sample, for comparison. The wind speed case study considers the effect of within-day variability, and min-max and mean approaches are applied to the 24 within-day hourly data samples.

The architecture of the NN model consists of one input, one hidden and one output layer. The number of input neurons is set to 2 for both case studies, since an auto-correlation analysis [43] has shown that the historical past values x_{t-1} and x_{t-2} should be used as input variables for predicting x_t in output. The number of hidden neurons is set to 10 for the synthetic case study and to 15 for the real case study, after a trial-and-error process. The number of output neurons is 1 in the input-interval case, since in this case a single neuron provides an interval in output; conversely, in order to estimate PIs starting from crisp input data, the number of output neurons must be set equal to 2, to provide the lower and upper bounds. As activation functions, the hyperbolic tangent function is used in the hidden layer and the logarithmic sigmoid function is used at the output layer. We remark that all arithmetic calculations throughout the estimation process of the interval-valued NN have been performed according to interval arithmetic (interval product, sum, etc.).

To account for the inherent randomness of NSGA-II, 5 different runs of this algorithm have been performed and an overall best non-dominated Pareto front has been obtained from the 5 individual fronts. To construct such best non-dominated front, the first (best) front of each of the 5 runs is collected, and the resulting set of solutions is subjected to the fast non-dominated sorting algorithm [41] with respect to the two objective functions. Then, the ranked non-dominated fronts $F_1, F_2, \dots, F_k$ are identified, where $F_1$is the best front, $F_2$is the second best front and F_k is the worst front. Solutions in the first (best) front F_1 are then retained as the overall best front solutions. This procedure gives us the overall best non-dominated Pareto front for the training set. After we have obtained this overall best front, we perform testing using each solution included in it.

For the firstcase study, the first 80% of the input data have been used for training and the rest for testing. For the second, a validation process has been performed. So the dataset has been divided into three parts: the first 60% is used for training, 20% for validation and the remaining 20% for testing. All data have been normalized within the range [0.1, 0.9].

Table 1 contains the parameters of the NSGA-II for training the NN. "MaxGen" indicates the maximum number of generations which is used as a termination condition andNc indicates the total number of individuals per population.$P_c$indicates the crossover probability and is fixed during the run.$P_{m_int}$is the initial mutation probability and it decreases at each iteration (generation) by the formula:

$$P_{m\_int} \times e^{(-\frac{gen}{MaxGen})} \tag{12}$$

TABLE I
NSGA-II and SOSA Parameters Used in the Experiments

Parameter	Numerical value
MaxGen	300
Nc	50
P_{m_int}	0.06
P_c	0.8
M	0.9
H	50
T_{init}	200
T_{min}	10^{-50}
CWC_{int}	10^{80}
Geometric cooling schedule of SA	$T_{k+1} = T_k * 0.95$

The average CPU times for both training and testing of NN have been recorded using MATLAB on a PC with 4 GB of RAM and a 2.53-GHz processor. The average CPU time for the entire training process with 300 generations takes around 5 hours; whereas the construction of testing PIs, i.e. for the online prediction of PIs, is very fast, of the order of1minute. It is needless to say that from the user point of view, the computational burden of the training phase is relatively less important [7], [8], since the training phase is, usually, only performed once. Note that computational load is dependent on the complexity of the structure of the model (e.g. number of input neurons, hidden layers, and hidden neurons), the size of the dataset and the performance of the learning algorithm. Moreover, as we have used MATLAB INTLAB Version 6 toolbox for all interval arithmetic calculations, the total computation time for training the NN with interval-valued

inputs has significantly increased compared to the CPU times of the case study performed in [11].

A. Synthetic Case Study

Four synthetic datasets have been generated according to the following model:

$$y(t) = f(t) + \delta(t), \tag{13}$$

where $f(t)$ is the deterministic component and $\delta(t)$ is the stochastic one, and the time horizon is 50 days which makes 1200 hours. The deterministic component has the following expression:

$$f(t) = 10 + 1.5 * sin\left(\frac{2\pi t}{T_1}\right) + sin\left(\frac{2\pi t}{T_2}\right), \tag{14}$$

where the period T_1 of the first periodic component has been set equal to 1 week, while T_2 is 1 day. The stochastic component $\delta(t)$of the generating model in (13) is given by an $ARMA(p, q)$ model [43], with $p = 2$ autoregressive terms, with same coefficients $\phi_1 = \phi_2 = 0.1$, and $q = 1$ innovation term with coefficient given by $\varphi_1 = 0.05$. Four different scenarios are then considered, which differ in the distribution chosen for the innovation term, and in the higher or lower innovation variability: in two of the four scenarios the innovation is Gaussian, and has variance equal to 1 and 9 respectively, while in the other two scenarios the innovation has a Chi-squared distribution, with 2 or 5 degrees of freedom (corresponding to a variance equal to 4 and 10, respectively). We thus generate four different 5-min time series datasets, from which we will obtain either crisp or interval hourly data.

Fig. 1 illustrates the testing solutions corresponding to the first (best) Pareto front found after training the NN on interval data constructed by the min-max approach (left) and mean approach (right). The plots show the solutions for the data generated from a Gaussian distribution. On each plot, two testing fronts are illustrated: the ones where solutions are marked as circleshave been obtained after training the NN on the interval data showing higher variability, while the ones with solutions marked as diamonds have been obtained after training the NN on the interval data having lower variability. Testing solutions obtained with data showing a lower variability are better than the ones with higher variability; hence, we can conclude that a higher variability in the input data may cause less reliable prediction results, and should thus be properly taken into account. Pareto fronts of solutions obtained for the data generated from a Chi-squared distribution are similar, and the results robust with respect to the choice of the innovation distribution.

Given the overall best Pareto set of optimal model solutions (i.e. optimal NN weights), it is necessary to select one NN model for use. For exemplification purposes, a solution is here subjectively chosen as a good compromise in terms of high PICP and low NMPIW. The selected solution is characterized by 95% CP and a NMPIW equal to 0.420 for the min-max approach applied to lower variability Gaussian data. The results on the testing set give a coverage probability of 95.5 % and an interval width of 0.412. Fig.2 shows 1-hour-ahead PIs for the selected Pareto solution, estimated on the testing set by the trained NN; the interval-valued targets included in the testing set are also shown in the figure.

Moreover, we also plot in Fig.3 the 5-min original time series data (testing set), corresponding to the generating scenario with Gaussian distribution and low variability, together with the estimated PIs corresponding to the selected solution: the solid line shows the 5-min original time series data, while the dashed lines are the PIs, estimated starting from interval input data constructed with the min-max approach within each hour. Since the time step for the estimated PIs is 1 hour, in order to compare them to the 5-min original time series data, we have shown in Fig.3 the same lower and upper bounds within each hour; thus, the PIs appear as a step function if compared to the original 5-min data.

INSERT FIGURE 1 (TWO COLUMNS)
INSERT FIGURE 2 (TWO COLUMNS)
INSERT FIGURE 3 (TWO COLUMNS)

In order to compare the Pareto front optimal solutions obtained with crisp and interval-valued inputs, a new normalized measure of the mean prediction interval width, named NMPIW*, has been a posteriori calculated as follows:

$$NMPIW^* = \frac{RT}{RRT} \times \frac{NRT}{0.8} \times NMPIW \tag{15}$$

where RT, RRT and NRT represent, respectively, the range of target (i.e., the range of the non-normalized hourly training data in input), the range of real target (i.e., the range of the non-normalized 5-min original time series data over the training set), and the range of normalized target (i.e., the range of the normalized hourly training data in input, $y_{max} - y_{min}$). Note that, unless the synthetic scenario changes, RRT takes the same value for min-max, mean and crisp approaches. The idea behind renormalization is to be able to compare PIs estimated from both interval and crisp approaches with respect to 5-min original time series data. As NMPIW for each solution on the Pareto front has been calculated by dividing the mean prediction interval width (MPIW) by the range of the training set in question, which is different for the two approaches, the Pareto fronts corresponding to the two approaches are not comparable. In order to analyze the performance of each approach with respect to 5-min original time series data, one should carry out a renormalization process which takes into account the range of the dataset involved in the comparison, and which leads the estimated PIs to a common unit of measure. As a numerical example for the calculation of NMPIW*, we have considered a testing solution, obtained on the synthetic data generated from the Gaussian distribution with lower variability and with the crisp approach, reported in Fig.4. The selected solution results in a coverage probability of 91% and an interval width of 0.328 on the testing.The values of RT, RRT and NRT are 6.87, 11.383,

and 0.647, respectively. Thus, by using (16), we have obtained NMPIW* as follows:

$$NMPIW^* = \frac{6.87}{11.383} \times \frac{0.647}{0.8} \times 0.328 = 0.16 \quad (16)$$

Moreover, for each solution on each Pareto front, a PICP* value has been a posteriori calculated. Equations (3) and (4) have been used with y_i representing non-normalized 5-min original time series data, and with $c_i = 1$, if $y_i \in [L(x_i), U(x_i)]$and otherwise$c_i = 0$, where $L(x_i)$and $U(x_i)$indicate de-normalized lower and upper bounds of the estimated PIs. Since estimated PIs have been obtained with hourly input data, while original data have a 5-min time frequency, in order to a posteriori calculate PICP* with respect to the original data we have assumed the same lower and upper bounds, $[L(x_i),U(x_i)]$, for each 5-min time step within each hour. Renormalization allows us to convert current Pareto fronts to new ones whose coverage probability and interval size are calculated according to the 5-min dataset, and are comparable across different (crisp and interval) approaches.

INSERT FIGURE 4 (TWO COLUMNS)

In Fig.4, a comparison between the testing fronts obtained with interval-valued and crisp inputs are illustrated. Solutions have been plotted according to the renormalized measures, i.e. the axes of the plots correspond to the new quantities NMPIW* and 1-PICP*, so that they can be compared. It can be appreciated that the solutions obtained with a crisp approach never result in coverage probabilities greater than 90% with respect to the original data. Furthermore, when the variability in the original data increases (right plots), the crisp approach gives less reliable results in terms of coverage probability, which is smaller than 80%.However, a model should take the within hour variability (high or low) into account and be capable of properly capturing it. Predictions resulting in a coverage probability lower than expected show the poor prediction power of the crisp approach, which cannot be considered a reliable support to decision making in the presence of high variability.

B. Real Case Study: Short-term Wind Speed Prediction

In this Section, results of the application of the proposed method to short-term wind speed forecasting with interval-input data are detailed. The dataset considered for the analysis consists in hourly wind speed data measured in Regina, Saskatchewan, a region of central Canada. Wind farms in Canada are currently responsible of an energy production of 5403 MW, a capacity big enough to power over 1 million homes and equivalent to about 2% of the total electricity demand in Canada [44]. The actual situation in Saskatchewan is characterized by the presence of 4 large wind farms located throughout the region, with a total capacity of approximately 198 MW [45].

The wind speed dataset, covering the period from January 1, 2010 till December 30, 2012, has been downloaded from the website [42]. Since hourly data have been collected, 24 wind speedvalues are available for each day. Fig.5 shows the behavior of hourly wind speed values only in the first 20 days, for the sake of clarity: one can appreciate the within-day variability in each individual day. The wind speed changes from 0 km/h to 72 km/h with an unstable behavior. From this raw hourly wind speed data, one can obtain daily interval wind speed data with the min-max and mean approach described at the beginning of Section IV. The so obtained datasetsinclude 1095 intervals among which the first 60% is used for training, 20% for validation and the remaining 20% for testing.

The procedure described in Sections II and IIIhas been applied for day-ahead wind speed prediction, both with interval and crisp inputs. Crisp results are reported for comparison, in terms of daily averages of the raw hourly data, with the same data splitting for training, validation and testing sets. The inputs are historical wind speed data $W_{t-1}$and W_{t-2}both for interval and crisp inputs; the optimal number of inputs has been chosen from an auto-correlation analysis [39].

When an optimal solution is selected from the front obtained by optimizing the NN on the basis of the training data, it is possible that the CP resulting from the application of this optimal NN to unseen data is lower than the one obtained on the training data.Thus,a validation set has beenalso selected, to test the generalization power of the proposed method. In other words, the aim is to test whether the selection of the solution with the required CP on the training data will result in well-calibrated PIs on the validation data or not. Fig. 6 showsthe values of PICP and NMPIW obtained on the validation set along the iterations of the MOGA (for the min-max approach). To obtain these graphs, at each iteration of the training process, we have selected the solution from the training front which either results in 90% PICP or is closest to (and, if possible, above) 90%. Then, the selected solution has been used on the validation set, and the corresponding PICP and NMPIW valueshavebeen recorded.The motivation behind these plots is to show the capability of the MOGA algorithmto generate reliable predictions on unseen data.

Table II reports the PICP and NMPIW values of the selected training and validation solutions corresponding to those having coverage probability between 90% and 100% on the overall best non-dominated Pareto front. These solutions are obtained by the min-max approach. From inspection both of Table II and the profiles of both objectives on the training and validation sets shown in Fig. 6, we can observe that the training, validation and testing results do not show significant difference. The PICP evaluation is coherent with NMPIW; hence, we can conclude that the proposed method results in well-calibrated PIs not only on the training set but also on the validation set.

In Fig. 7, the testing solutions obtained with the interval-valued min-max and mean approaches, and with crisp inputs, are illustrated. The figure has been plotted according to the renormalized solutions, as explained in Section IV-A, i.e. the axes of the plot correspond to the new quantities NMPIW* and 1-PICP*. As already appreciated in the synthetic case study, one can notice that the solutions obtained with a crisp

approach do not result in a coverage probability larger than 95% with respect to the original data. Furthermore, looking at the solutions in Fig. 7 which show a CP greater than 90%, the ones corresponding to the crisp approach give larger interval size. Since in practice it is important to have narrow PIs with high coverage probability, an interval-inputs approach is more suited to reliable decision making.

TABLE II
TRAINING, VALIDATION AND TESTING RESULTS OBTAINED BY NSGA-II

Training		Validation		Testing	
PICP (%)	NMPIW	PICP (%)	NMPIW	PICP (%)	NMPIW
90.1	0.440	91.0	0.470	91.4	0.452
90.3	0.446	92.0	0.474	92.6	0.461
90.6	0.452	91.6	0.482	92.6	0.466
91.6	0.456	92.5	0.486	93.3	0.471
92.1	0.466	93.1	0.494	93.9	0.480
93.2	0.487	94.3	0.514	95.1	0.500
93.6	0.493	94.4	0.526	94.8	0.506
94.3	0.529	96.0	0.562	96.6	0.546
96.9	0.578	97.7	0.606	98.3	0.595
97.9	0.636	98.9	0.673	98.9	0.651
98.5	0.662	99.2	0.692	99.3	0.679
99.2	0.721	99.5	0.757	99.7	0.739

INSERT FIGURE 5 (TWO COLUMNS)
INSERT FIGURE 6 (TWO COLUMNS)
INSERT FIGURE 7 (TWO COLUMNS)

From the overall best Pareto set of optimal solutions (i.e. optimal NN weights) obtained after training the network on the interval input data constructed with the min-max and mean approaches, a solution must be chosen. The selection of the solution might be accomplished by setting a constraint on one of the objective and choosing the optimal value for the other one, or by considering some other methods to weigh the two objectives [46]. In general, the selection should represent the preferences of the decision makers (DMs). Here, for simplicity's sake, we do not introduce any specific formal method of preference assignment but subjectively choose a good compromise solution: for the min-max approach, the results give a coverage probability of 92.1% and interval width of 0.466 on the training, and a coverage probability of 93.9% and interval width of 0.480 on the testing. For the mean approach, the selected solution results in a coverage probability of 91.7% and interval width of 0.424 on thetraining, and a coverage probability of 93% and interval width of 0.437on the testing.

Figs.8 and 9 reportday-ahead PIs (dashed lines) for the selected Pareto solutions, with respect to the mean and min-max approaches respectively, estimated on the testing set by the trained NN.The interval-valued targets (solid lines) included in the testing set are also shown in the figures. As wind speed cannot be negative, to reflect the real physical phenomena the negative lower bounds of the PIs have been replaced with zeros. From inspection of the figures, we observe that the target profile of the mean approach is more accurate if compared to that of the min-max approach. However, the peak points have beencovered relatively better by the min-max approach if compared to the mean. Hence, which one would be preferably chosen depends on the application.The mean approach might be considered more similar to classical methods for short-term wind speed/power prediction using single-valued data as inputs, obtained as a within-hour or within-day average. By this approach we canadd information tothe single-valuedaverages, and thus wecan include in the model the potential uncertainty caused by the data itselfshowing a within hour/day variability.Hence, the mean approachis a well-suited interval inputs alternative to the classical crisp inputs one, and it might be considered more feasible in practice.

INSERT FIGURE 8 (TWO COLUMNS)
INSERT FIGURE 9 (TWO COLUMNS)

In order to compare the interval-valued and crisp approaches in a clear way, we have shown the PIs obtained by both approaches in one Figure(see Figs. 10 and 11). In Fig. 10, we have shown the estimated day-ahead PIs corresponding to the selected solutions obtained by mean and crisp approaches, respectively, on the daily crisp wind speed testing set by the trained NN. The solutions have been selected from the overall best Pareto set of optimal solutions obtained by mean and crisp approaches. These solutions result in 91.8% CP* and 0.483 NMPIW* for the mean approach, and has 91.3% CP and 0.495 NMPIW for the crisp approach, on the testing dataset. It is clear that the solution obtained by the mean approach dominates the one obtained by the crisp approach. Note that PICP* and NMPIW* values have been a posteriori calculated only for the mean approach; as thecrisp approach has been trained with the crisp daily wind speed training set, it is not necessary to convert PICP and NMPIW to PICP* and NMPIW* values.

Similarly, Fig. 11 has been plotted by considering a posteriori calculated PICP* and NMPIW* values (see Fig. 7) corresponding to thetwo solutions selected from the overall best Pareto fronts of min-max and crisp approaches, respectively. These solutions result in 91.4% CP* and 0.452 NMPIW* for min-max approach, and 91.2 % CP* with 0.472 NMPIW* for crisp approach, on the testing dataset (raw hourly wind speed data). It is obvious that the solution obtained by min-max approach is superior to the one obtained by crisp approach. In other words, we have obtained higher quality PIs with interval-valued input approach. Note that this comparison is done on the raw hourly wind speed dataset. Since the time step for the estimated PIs is 1 day, in order to compare them to the hourly original time series data, we have shown in Fig. 11 the same lower and upper bounds within each day; thus, the PIs appear as a step function if compared to the

original 1-hour data. Due to space limitations we have only plotted the estimated PIs obtained by min-max approach.

TABLE III
PICP AND NMPIW VALUES OBTAINED BY SOSA WITH RESPECT TO WIND SPEED DATASET (TRAINING / TESTING)

SOSA METHOD	PICP (%)	NMPIW	CWC
1	93.8 / 95.6	0.567 / 0.578	0.649 / 0.578
2	71.7 / 73.8	0.300 / 0.312	2897 / 1032
3	72.0 / 75.2	0.297 / 0.310	2425 / 519.6
4	75.5 / 76.3	0.317 / 0.328	439.0 / 311.3
5	92.1 / 95.1	0.725 / 0.752	0.978 / 0.752

INSERT FIGURE 10 (TWO COLUMNS)
INSERT FIGURE 11 (TWO COLUMNS)

From the results illustrated in Figs. 10 and 11, one might comment that the PIs obtained with the interval inputs approach are capable of capturing the peak points (highest and lowest) of the target of interest (hourly data). Although there are some highly extreme values dropping out of the estimated PIs, the interval approach leads to better coverage of the intermittent characteristic of wind speed than the crisp approach. In other words, the interval approach manages to describe more efficiently the short-term variability of wind speed.

C. Comparison with single-objective simulated annealing (SOSA) method

In this section, we present the results from a comparison with a method called "Lower and Upper Bound Estimation (LUBE)" proposed by Khosravi et al. in [7] to estimate PIs with single-valued (crisp) inputs. In their paper, the authorshave used single-objective simulated annealing algorithm (SOSA) to train the NN and adopted the cost function defined in (17),which combines PICP and NMPIW, to be minimized.

The cost function proposed in [7] is called coverage width-based criterion (CWC):

$$CWC = NMPIW(1 + \gamma(PICP)e^{-\eta(PICP-\mu)}) \qquad (17)$$

whereη and μ are constants. The role of η is to magnify any small difference between μ and PICP. The value of μ gives the nominal confidence level, which is set to 90% in our experiments (see Table I). Then, η and μ are two parameters determining how much penalty is paid by the PIs with low coverage probability. The function $\gamma(PICP)$ is equal to 1 during training, whereas in the testing of the NN is given by the following step function:

$$\gamma(PICP) = \begin{matrix} 0, & PICP \geq \mu \\ 1, & PICP < \mu \end{matrix} \qquad (18)$$

To perform a comparison between SOSA and the proposed MOGA method, we have run the SOSA by using the same interval-valued wind speed training data. For SOSA, the initial temperature has been determined after a trial and error procedure. It has been tried with values of 5, 200 and 500: it turns out that the SOSA with initial temperature of 200 gives best performance.Table I contains the parameters of the SOSA;the maximum number of generation has been set to 500.

The training process has been repeated five times. Training and testing results in each run have been reported in Table III.Due to space limitation, we have put only min-max approach results.

According to the results reported in Table III, it can be observed that the training and corresponding testing solutions do not show high consistency in terms of coverage probability and interval size among the five runs performed. In other words, there is a high difference among the results: SOSA gives high CP value in one run whereas it generates less accurate PIs in another one:3 out of 5 runs give CP values smaller than the predetermined nominal confidence level, i.e. 90% in our experiments. Thus CWC values are quite high for those runs. Although the existing works done based on the SOSA LUBE method with single-valued inputsshow promising results for the construction of PIs [7]-[10], the reported results in the present work (see Table III) demonstrate a drawback about SOSA method's robustness on this specific problem.

For comparison purpose, we haveselected the run giving the smallest CWC value on the training set, which is 0.649.Note that in previous works of literature [7], [47], the mean or median value of several runs has been used as final prediction result. The selected run results in 93.8% CP and 0.567 NMPIW on the training set, and a coverage probability of 95.6% and interval width of 0.578 on the testing. By comparison, we have selected a solution from the overall best Pareto front obtained by MOGA min-max approach. This selected solutiongives a coverage probability of 94.3 % and interval width of 0.529 on the training, and a coverage probability of 96.6 % and interval width of 0.546 on the testing.For what concerns the mean approach, we have observed similar results: 2 out of the 5 runs have given CP less than 90% both on training and testing sets. The runs resulting in coverage probability bigger than 90% have quite large interval widths (above 50%). We have selected a run which has the smallest CWC value: it has a coverage probability of 93.1% with 0.520 NMPIW on the training and 94.7% CP and interval width of0.531 on the testing datasets. On the contrary, the MOGA method has given a solution with 93.3% CP with 0.440 interval size on the training, and 94.4% CP with 0.453 interval size on the testing set.

It is clear that the solutions obtained by MOGA dominate the best ones obtained by SOSA. It is worth pointing out that as both solutions obtained by min-max method give large interval sizes (around 50%) they cannot provide useful information in practice, because the uncertainty level is too high to support a reliable and informed decision in typical application contexts. However, with the MOGA approach one can select a solution from the Pareto frontgiving tight PIW with a high CP, which satisfies the predetermined nominal

confidence level. In short, from the results reported in Table III, one can conclude that the SOSA method does not give high quality PIs with respect to the interval-valued time series forecasting case study considered in this work.

V. Conclusions

The goal of the research presented in this paper is to quantitatively represent the uncertainty in neural networks predictions of time series data, originating both from variability in the input and in the prediction model itself. The application focus has been on wind speed, whose forecasting is crucial for the energy market, system adequacy and service quality in power grid with integrated wind energy systems. Accuracy of predictions of power supply and quantitative information on the related uncertainty is relevant both for the power providers and the system operators.

Specifically, we have presented two approaches that can be used to process interval-valued inputs with multi-layer perceptron neural networks. The method has been applied on a synthetic case study and on a real case study, in which the data show a high (short-term) variability (within hour and within day). The results obtained reveal that the interval-valued input approach is capable of capturing the variability in the input data with the required coverage. The results enable different strategies to be planned according to the range of possible outcomes within the interval forecast.

As for future research, the use of an ensemble of different NNs will be considered to further increase the accuracy of the predictions, andtype-2 fuzzy setscan be integrated into the proposed model as an alternative way to represent the input uncertainty.

References

[1] J. C. Refsgaard, J. P. van der Sluijs, J. Brown, and P. van der Keur, "A framework for dealing with uncertainty due to model structure error,"*Advances in Water Resources*, vol. 29, no. 11, pp. 1586-1597, Nov. 2006.

[2] H. Cheng, "Uncertainty quantification and uncertainty reduction techniques for large-scale simulations," Ph.D. dissertation, Virginia Polytechnic Institute and State University, Virginia, 2009.

[3] E. Zio and T. Aven, "Uncertainties in smart grids behavior and modeling: What are the risks and vulnerabilities? How to analyze them?," *Energy Policy*, vol. 39, no. 10, pp. 6308-6320, Oct. 2011.

[4] N. Pedroni, E. Zio, and G. E. Apostolakis, "Comparison of bootstrapped artificial neural networks and quadratic response surfaces for the estimation of the functional failure probability of a thermal–hydraulic passive system", *Reliability Engineering and System Safety*, vol. 95, no. 4, pp. 386-395, Apr. 2010.

[5] H. Agarwal, J. E. Renaud, E. L. Preston, and D. Padmanabhan, "Uncertainty quantification using evidence theory in multidisciplinary design optimization,"*Reliability Engineering and System Safety*, vol. 85, no. 1-3, pp. 281-294, July-Sep. 2004.

[6] J. C. Helton, "Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty,"*Journal of Statistical Computation and Simulation*, vol. 57, no. 1-4, pp. 3-76, 1997.

[7] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower Upper Bound Estimation Method for Construction of Neural Network-Based Prediction Intervals," *IEEE Transactions on Neural Networks*, vol. 22, no. 3, pp. 337-346, March 2011.

[8] H. Quan, D. Srinivasan, and A. Khosravi. "Short-Term Load and Wind Power Forecasting Using Neural Network-Based Prediction Intervals," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 303-315, Feb. 2014.

[9] A. Khosravi, S. Nahavandi, and D. Creighton, "A prediction interval-based approach to determine optimal structures of neural network metamodels," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2377-2387, March 2010.

[10] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances," *IEEE Transactions on Neural Networks*, vol. 22, no. 9, pp. 1341-1356, Sept. 2011.

[11] R. Ak, Y-F. Li, V. Vitelli, and E. Zio, "Multi-objective Genetic Algorithm Optimization of a Neural Network for Estimating Wind Speed Prediction Intervals," Applied Soft Computing (resubmitted), 2014.

[12] H. Papadopoulos, V. Vovk, and A. Gammerman, "Regression Conformal Prediction with Nearest Neighbours," *Journal of Artificial Intelligence Research*, vol. 40, pp. 815-840, 2011.

[13] H. Papadopoulos and H. Haralambous,"Reliable Prediction Intervals with Regression Neural Networks,"*Neural Networks*, vol. 24, no. 8, pp. 842-851, 2011.

[14] L. V. Barboza, G. P. Dimuro, and R. H. S. Reiser, "Towards interval analysis of the load uncertainty in power electric systems,"*in Proc. International Conference on Probabilistic Methods Applied to Power Systems*, 2004, pp. 538-544.

[15] A. M. Roque, C. Maté, J. Arroyo, and Á. Sarabia, "iMLP: Applying Multi-Layer Perceptrons to Interval-Valued Data," *Neural Processing Letters*, vol. 25, no. 2, pp. 157–169, Apr. 2007.

[16] A. L. S. Maia, F. A .T. de Carvalho, and T. B. Ludermir, "Forecasting models for interval-valued time series," *Neurocomputing*, vol. 71, no. 16, pp. 3344-3352, 2008.

[17] H. J. Zimmermann, Fuzzy Set Theory-And Its Applications.4th ed., USA: Kluwer Academic Publishers, 2001, pp. 1-514.

[18] D. Zhai and J. M. Mendel, "Uncertainty measures for general Type-2 fuzzy sets," Information Sciences, vol. 181, no. 3, pp. 503-518, Feb. 2011.

[19] L. Qilian and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 535-550, 2000.

[20] F. Gaxiola, P. Melin, F. Valdez, and O. Castillo, "Interval type-2 fuzzy weight adjustment for backpropagation neural networks with application in time series prediction," *Information Sciences*, vol. 260, pp. 1-14, Mar. 2014.

[21] J. A. Sanz, A. Fernandez, H. Bustince, and F. Herrera, "IVTURS: A Linguistic Fuzzy Rule-Based Classification System Based On a New Interval-Valued Fuzzy Reasoning Method With Tuning and Rule Selection," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 399-411, June 2013.

[22] P. Sussner, M. Nachtegael, T. Mélange, G. Deschrijver, E. Esmi, and E. Kerre, "Interval-Valued and Intuitionistic Fuzzy Mathematical Morphologies as Special Cases of L-Fuzzy Mathematical Morphology," *Journal of Mathematical Imaging and Vision*, vol. 43, no. 1, pp. 50-71, May 2012.

[23] M.Nachtegael, P.Sussner, T. Mélange, and E. E.Kerre,"On the role of complete lattices in mathematical morphology: From tool to uncertainty model," *Information Sciences*, vol. 181, no. 10, pp. 1971-1988, 2011.

[24] E. Hofer, M.Kloos, B.Krzykacz-Hausmann, J.Peschke, and M.Woltereck, "An approximate epistemic uncertainty analysis approach in the presence of epistemic and aleatory uncertainties", *Reliability Engineering and System Safety*, vol. 77, no. 3, pp. 229-238, Sep. 2002.

[25] J. C. Helton and F. J. Davis, "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems,"*Reliability Engineering and System Safety*, vol. 81, no 1, pp. 23-69, July 2003.

[26] R. E. Moore, R. B. Kearfott, and M. J. Cloud.Introduction to Interval Analysis, Society for Industrial and Applied Mathematics. 1st ed., USA, 2009, pp. 1-235.

[27] J. Arroyo and C. Maté,"Introducing interval time series: Accuracy measures,"*in Proc. COMPSTAT*, 2006, pp. 1139-1146.

[28] World Wind Energy Association, Half Year Report. Oct. 2013, 1-22. http://www.wwindea.org/webimages/WorldWindEnergyReport2012_final.pdf; [Accessed on October 2013].

[29] A. U.Haque, P. Mandal, M. E. Kaye, J. Meng, L. Chang, and T. Senjyu,"A new strategy for predicting short-term wind speed using soft computing models," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 7, pp. 4563-4573, 2012.

[30] J.Jaesungand R. P. Broadwater, "Current status and future advances for wind speed and power forecasting," *Renewable and Sustainable Energy Reviews*, vol. 31,pp. 762-777, March 2014.
[31] H. Bouzgou and N. Benoudjit, "Multiple architecture system for wind speed prediction," *Applied Energy*, vol. 88, no. 7, pp. 2463-2471, 2011.
[32] A. More and M. C. Deo, "Forecasting wind with neural networks,"*Marine Structures*, vol. 16, no. 1, pp. 35-49, Jan. 2003.
[33] R. Ak, Y. F. Li, V. Vitelli, E. Zio, E. López Droguett, and C. MagnoCouto Jacinto, "NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1205-1212, March 2013.
[34] M. Pulido, P. Melin, and O. Castillo, "Genetic optimization of ensemble neural networks for complex time series prediction,"*in Proc. IJCNN*, pp. 202-206, July 31- Aug. 5 2011.
[35] E. Zio, "A study of the bootstrap method for estimating the accuracy of artificial neural networks in predicting nuclear transient processes,"*IEEE Transactions on Nuclear Science*, vol. 53, no. 3, pp. 1460-1478, June 2006.
[36] D. L. Shrestha and D. P. Solomatine, "Machine learning approaches for estimation of prediction interval for the model output," *Neural Networks*, vol. 19, no. 2, pp. 225-235, 2006.
[37] Y. Sawaragi, H. Nakayama, and T. Tanino.Theory of Multiobjective Optimization. Orlando, FL: Academic Press Inc., 1985, pp. 1-296.
[38] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial,"*Reliability Engineering and System Safety*, vol. 91, no. 9, pp. 992-1007, Sep. 2006.
[39] E. Zitzlerand L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach,"*IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, Nov. 1999.
[40] N. Srinivas and K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.
[41] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II ,"*IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, Apr. 2002.
[42] Canadian Weather Office, 2012. Available: http://www.weatheroffice.gc.ca/canada_e.html. [Accessed: 08-avr-2013].
[43] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, Time Series Analysis: Forecasting and Control.4th ed., Wiley, 2008.
[44] Media Kit 2012, Canadian Wind Energy Association (CanWEA).Available: http://www.canwea.ca/pdf/windsight/CanWEA_MediaKit.pdf
[45] SaskPower Annual Report 2011. Available: http://www.saskpower.com/news_publications/assets/annual_reports/2011_skpower_annual_report.pdf
[46] E. Zio, P. Baraldi, and N. Pedroni, "Optimal power system generation scheduling by multi-objective genetic algorithms with preferences," *Reliability Engineering and System Safety*, vol. 94, pp. 432-444, 2009.
[47] A. Khosravi and S. Nahavandi, "Combined Nonparametric Prediction Intervals for Wind Power Generation," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 4, pp. 849‑856, Oct. 2013.

Ronay Ak received the B.Sc. degree in mathematical engineering from Yildiz Technical University in 2004, the M.Sc. degree in industrial engineering from Istanbul Technical University, Turkey in 2009, and Ph.D. degreein Energy specialty at Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Électricité de France (EDF), ÉcoleCentrale Paris (ECP) and ÉcoleSuperieured'Électricité (SUPELEC), France, in July 2014. Her research interests include uncertainty quantification, prediction methods, artificial intelligence, reliability analysis of wind-integrated power networks, and multi-objective optimization.

Valeria Vitellireceived the Ph.D. degree in Mathematical Models and Methods for Engineering, with a focus on statistical models for classification of high-dimensional data, in May 2012.She worked as a postdoc researcher within the Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy- Électricité de France (EDF), ÉcoleCentrale Paris (ECP) and ÉcoleSupérieured'Électricité (SUPELEC), France, from February 2012 to May 2013. She is currently a postdoc researcher in the Department of Biostatistics, University of Oslo, Norway. Her current research interests concern prediction and uncertainty quantification methods for complex data.

Enrico Zio (M,SM**)** received the Ph.D. degree in nuclear engineering from Politecnico di Milano and MIT in 1995 and 1998, respectively. He is currently Director of the Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy- Électricité de France (EDF), at ÉcoleCentrale Paris (ECP) and ÉcoleSuperieured'Électricité (SUPELEC) and full professor at Politecnico di Milano. His research focuses on the characterization and modeling of the failure/repair/maintenance behavior of components, complex systems and their reliability, maintainability, prognostics, safety, vulnerability and security, Monte Carlo simulation methods, soft computing techniques, and optimization heuristics.

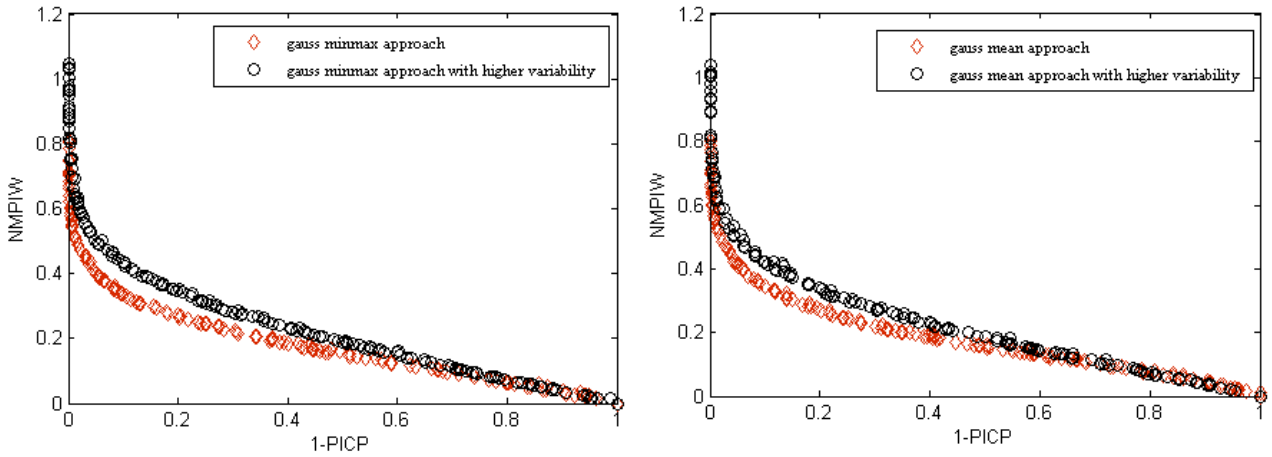

Fig. 1.Testing solutions for the Gaussian time series: min-max approach (left) and mean approach (right).

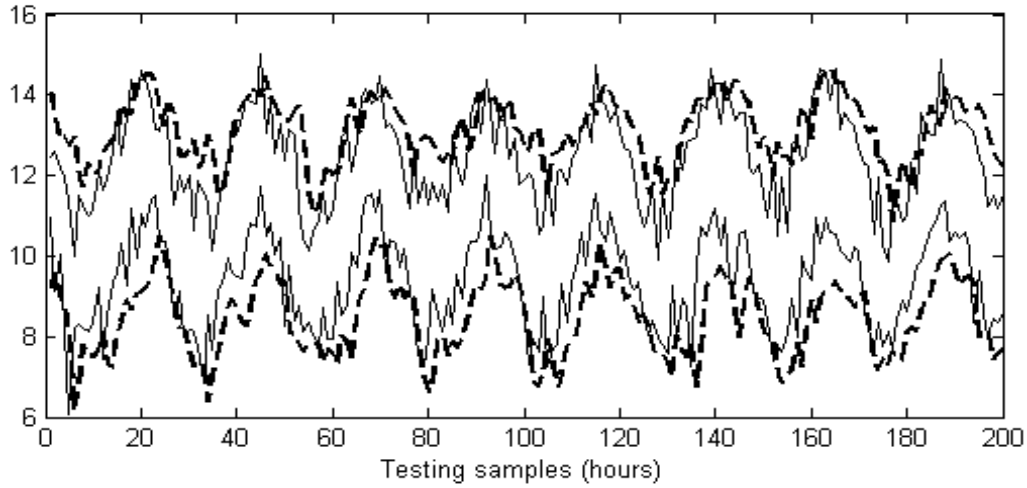

Fig. 2.Estimated PIs for 1-h ahead prediction on the testing set (dashed lines), and interval-valued input data (target) constructed by the min-max approach from the Gaussian distribution scenario with lower variability on the testing set (solid lines).

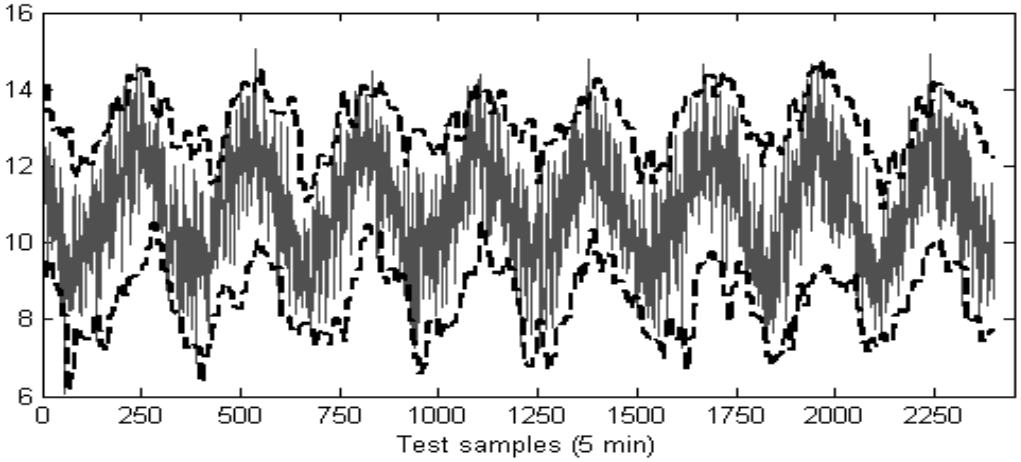

Fig. 3.Estimated PIs for 1-h ahead prediction on the testing set (dashed lines), and the original 5-min time series data on the testing set (solid line) obtained in the Gaussian distribution scenario with lower variability.

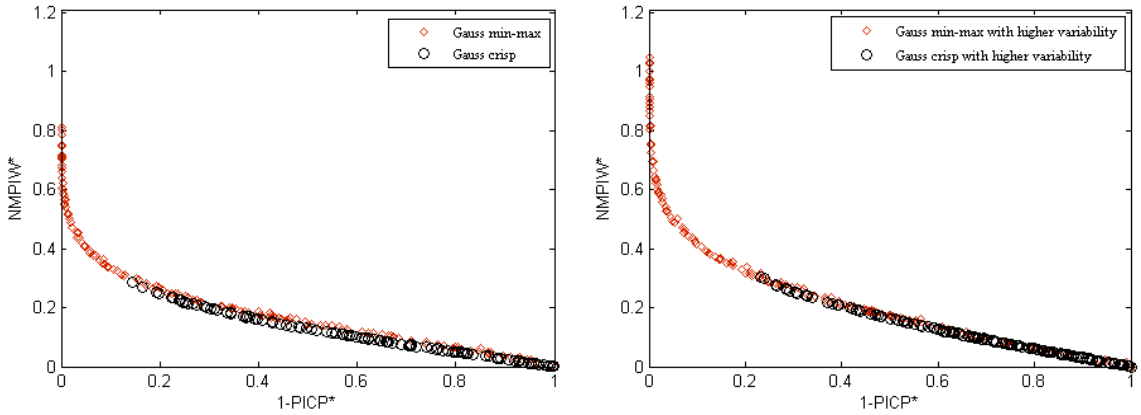

Fig. 4.Testing solutions obtained in the synthetic case study with interval-valued (min-max approach) and crisp approaches: data have been generated from the Gaussian distribution with lower (left) and higher variability (right).

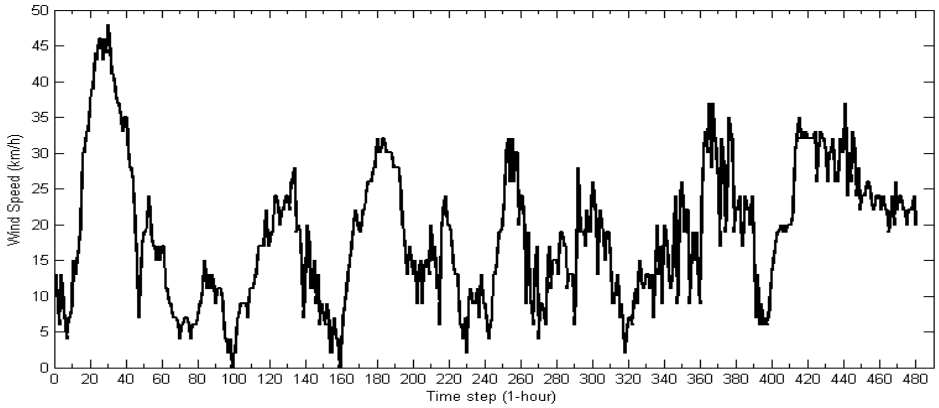

Fig. 5.The raw hourly wind speed dataset used in this study: first 20 days.

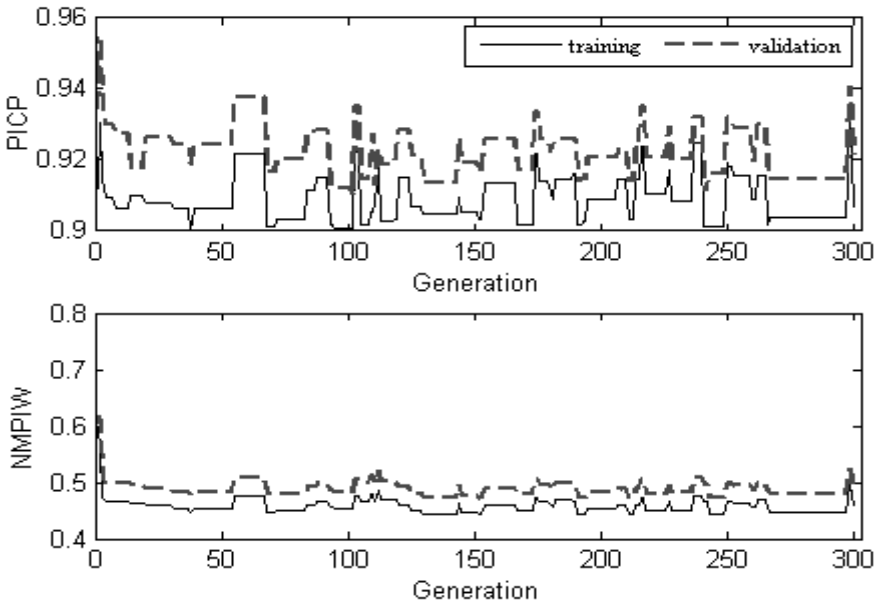

Fig. 6.Evaluation of PICP (top) and NMPIW (bottom) with respect to training and validation setsalong MOGA iterations, considering interval inputs obtained with a min-max approach.

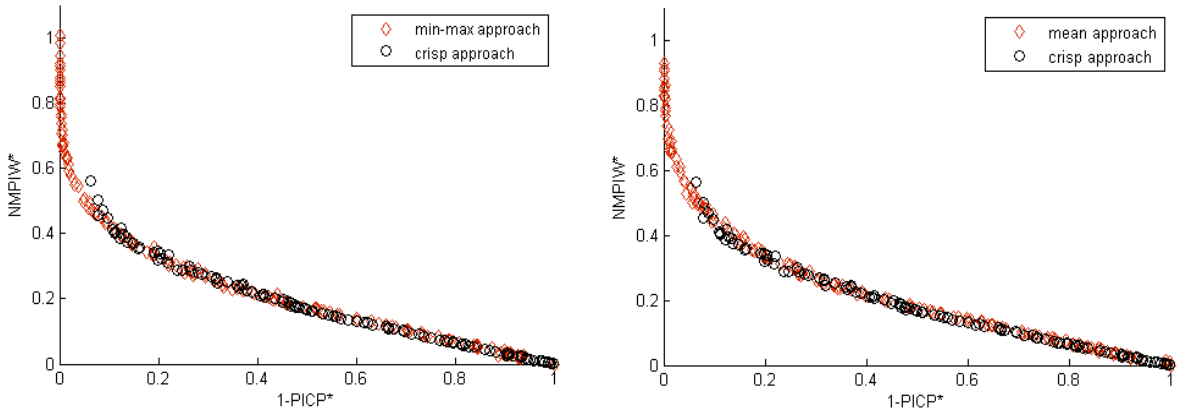

Fig. 7.Comparison between crisp and interval-valued approaches testing solutions, after renormalization, for day-ahead wind speed prediction: min-max with respect to crisp approach comparison (left), and mean with respect to crisp approach comparison (right).

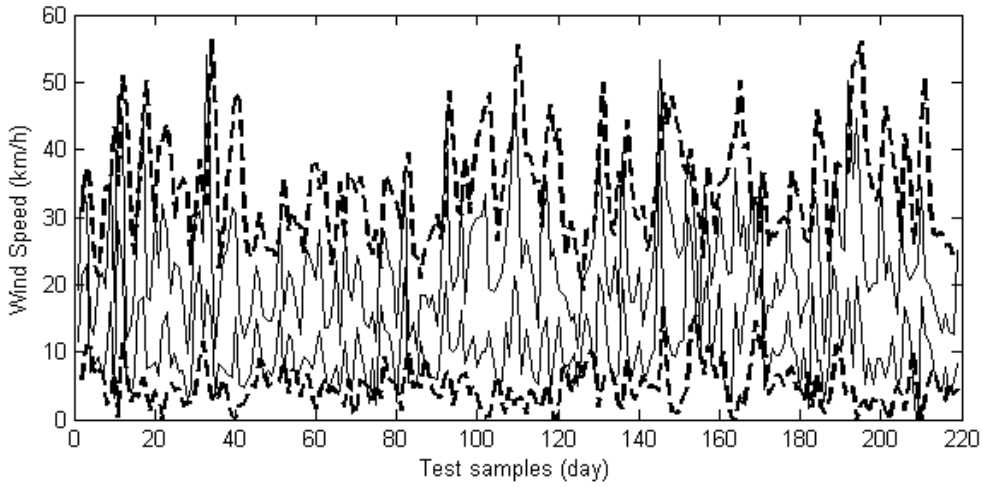

Fig. 8.Estimated PIs with interval inputs for day-ahead wind speed prediction on the testing set (dashed lines), and interval-valued wind speed data (constructed by the mean approach) included in the testing set (solid line).

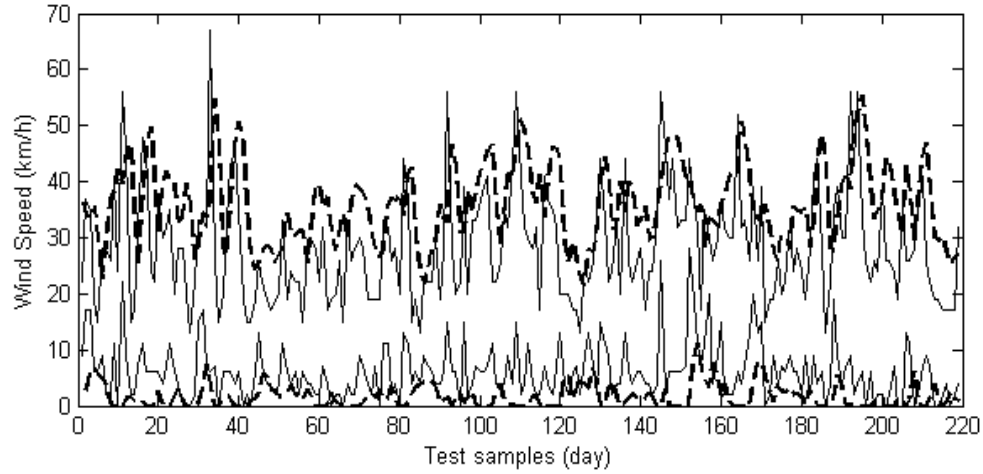

Fig. 9.Estimated PIs (dashed lines) with interval inputs for day-ahead wind speed prediction on the testing set and interval-valued wind speed data (constructed by the min-max approach) included in the testing set (solid line).

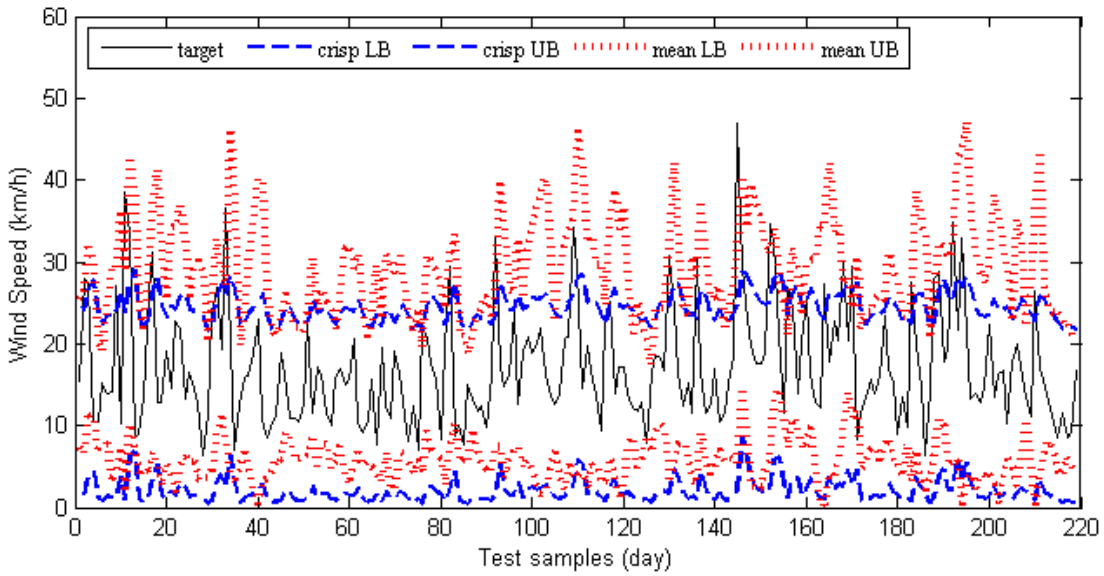

Fig. 10.Estimated PIs with interval (dotted red lines) and crisp (dashed blue lines) inputs for day-ahead wind speed prediction on the testing set and single-valued (crisp) daily wind speed data included in the testing set (solid line).

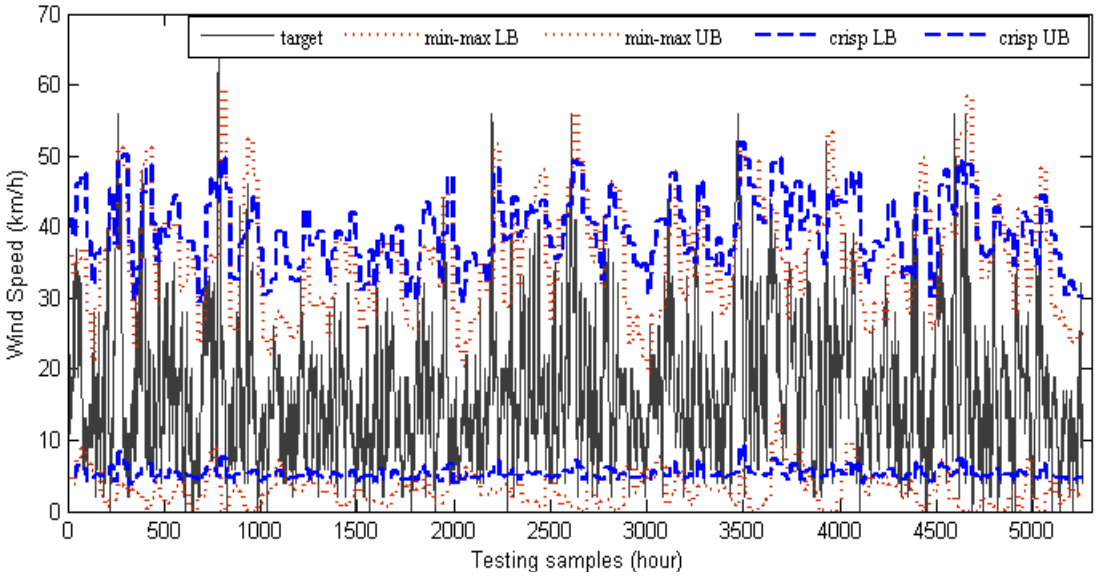

Fig. 11. Estimated PIs with interval (dotted red lines) and crisp (dashed blue lines) inputs for day-ahead wind speed prediction on the testing set and single-valued (crisp) raw hourly wind speed data (solid line).