IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 5, 2021, accepted January 21, 2021, date of publication February 1, 2021, date of current version February 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056003

Review on Fuzzy and Neural Prediction Interval Modelling for Nonlinear Dynamical Systems

OSCAR CARTAGENA[1], (Graduate Student Member, IEEE), SEBASTIÁN PARRA[1], (Member, IEEE), DIEGO MUÑOZ-CARPINTERO[2], (Member, IEEE), LUIS G. MARÍN[3], (Member, IEEE), AND DORIS SÁEZ[1,4], (Senior Member, IEEE)

[1]Department of Electrical Engineering, Faculty of Mathematical and Physical Sciences, University of Chile, Santiago 8370451, Chile
[2]Institute of Engineering Sciences, Universidad de O'Higgins, Rancagua 2841935, Chile
[3]Department of Electrical and Electronics Engineering, Universidad de Los Andes, Bogotá 111711, Colombia
[4]Instituto Sistemas Complejos de Ingeniería, Santiago 8370451, Chile

Corresponding author: Doris Sáez (dsaez@ing.uchile.cl)

This study was supported by the Fondo Nacional de Desarrollo Cientìfico y Tecnològico (FONDECYT) under grant 1170683, the Solar Energy Research Center SERC-Chile under grant ANID/FONDAP/15110019, Instituto Sistemas Complejos de Ingeniería (ISCI) under grant ANID PIA/BASAL AFB180003 and ANID/PAI Convocatoria Nacional Subvención a Instalación en la Academia Convocatoria 2019 PAI77190021. The work of Oscar Cartagena was supported by the ANID-PFCHA/Doctorado Nacional/2020-21200709.

ABSTRACT The existing uncertainties during the operation of processes could strongly affect the performance of forecasting systems, control strategies and fault detection systems when they are not considered in the design. Because of that, the study of uncertainty quantification has gained more attention among the researchers during past decades. From this field of study, the prediction intervals arise as one of the techniques most used in literature to represent the effect of uncertainty over the future process behavior. Thus, researchers have focused on developing prediction intervals based on the use of fuzzy systems and neural networks, thanks to their usefulness for represent a wide range of processes as universal approximators. In this work, a review of the state-of-the-art of methodologies for prediction interval modelling based on fuzzy systems and neural networks is presented. The main characteristics of each method for prediction interval construction are presented and some recommendations are given for selecting the most appropriate method for specific applications. To illustrate the advantages of these methodologies, a comparative analysis of selected methods of prediction intervals is presented, using a benchmark series and real data from solar power generation of a microgrid.

INDEX TERMS Prediction intervals, fuzzy interval, neural network intervals, uncertainty.

I. INTRODUCTION

The use of fuzzy logic systems (FLS) and neural networks (NNs) has proliferated in the literature for the modeling of systems and time series. Since both types of models are known as universal approximators that can identify relationships between inputs and target (outputs) variables, they are generally used when the system or the time series to be modeled follows nonlinear dynamics [1]. Although FLS and NN exhibit adequate performance to obtain the expected value of these kind of systems, uncertainty is not typically quantified by these modelling approaches. However, information on the dispersion of the output of the model provides more information about the phenomena. and more useful information from a decision-making point of view than the models with only expected values [2], [3].

Prediction intervals have been proposed to address the problematic of quantifying prediction uncertainty. A prediction interval establishes a range around the output of the model, representing the uncertainty present in the system. The main motivation for the construction of prediction intervals is to quantify the uncertainty in the point prediction to provide information about the uncertainty and enable the consideration of multiple scenarios for the best and worst conditions of the system [4].

Usually the width of the prediction interval increases with the prediction horizon due to the coupling effect and propagation of the different sources of uncertainty in the model. In several applications, such as the implementation of forecasting systems, control strategies, fault detection methods

The associate editor coordinating the review of this manuscript and approving it for publication was Valentina E. Balas.

or decision support systems, where decisions are taken based on the predicted future behavior of the modeled system, it is desired that the prediction intervals are as precise as possible. Therefore, the aim to prediction interval construction is to find the narrowest intervals possible that contain a desired percentage of measured data over the future time horizon.

Considering the importance of FLS and NN for the modeling of nonlinear systems and time series, several methods have been proposed in order to construct an appropriate prediction interval based on these models. The main contribution of this paper is to analyze and compare features and the performance of the methods for construction of prediction intervals based on FLS and NN reported in the specialized literature. In order to evaluate the prediction intervals performance, several methods are compared based on two dynamical nonlinear systems implemented: modified Chen series and solar power generation data from Milan. Finally, the characteristics and applicability of each discussed methods are presented.

The remainder of this paper is organized as follows: Section II presents the fundamentals of prediction intervals. Sequential and direct methods for both fuzzy and neural prediction intervals construction are described in section III and IV respectively. The characteristics and a discussion of the methods are presented in section V. Section VI presents the results of a benchmark test and case study from solar power generation of a microgrid. The last section provides the main conclusions.

II. FUNDAMENTALS OF PREDICTION INTERVALS

A. SOURCES OF UNCERTAINTY IN SYSTEM MODELING

In order to fully understand the information provided by Prediction Intervals, it is important to study what are the possible causes of uncertainty in dynamical system modeling, and on what phenomena they can originate.

In theory, predictive modeling assumes that all observations are generated by a data generating function $f(x)$ combined with and additive noise [4]–[6]

$$y = f(x) + \epsilon, \tag{1}$$

where ϵ is a zero mean random variable called data noise that is responsible for introducing uncertainty modeling into predictive models in the form of **aleatory uncertainty**, whose origin can be traced to the exclusion of complicated variables from the model which cannot be determined with sufficient precision, or due to the presence of inherently stochastic processes in the observed system or data obtention procedure.

Based on this formulation, predictive models attempt to produce an estimate $\hat{f}(x)$ of the data generating function in order to calculate predictions of the expected value of the system. The model of $\hat{f}(x)$ is often referred to as crisp model. This procedure introduces an additional form of uncertainty, known as **epistemic uncertainty**, since the crisp model is only an approximation of the true data generating function. Assuming both types of uncertainty are independent, the total variance of observations can be expressed as

$$\sigma^2_{total} = \sigma^2_{model} + \sigma^2_{data} \tag{2}$$

where σ^2_{model} is attributed to epistemic uncertainty and σ^2_{data} is attributed to aleatory uncertainty.

Since the factors contributing to epistemic uncertainty can vary greatly, some authors [7] have proposed subsequent classifications for this term:

1) **Model misspecification:** Uncertainty determined by how close the estimate $\hat{f}(x)$ can approximate the real data generating function $f(x)$ under optimal parameter and data conditions
2) **Training data uncertainty:** Uncertainty over how representative the training data is with respect to the whole input distribution, and how sensitive the model can be to unseen samples
3) **Parameter uncertainty:** Uncertainty on the values of the model parameters due to local minima stagnation or premature termination.

It is important to note that since total uncertainty can come from many diverse sources, the expression can be highly complex and difficult to quantify, which is why interval modeling has been proposed as a solution to this problem.

B. TYPES OF INTERVALS AND PREDICTION INTERVAL METHODS

Regarding to interval models it is important to differentiate confidence intervals from prediction intervals.

Strictly speaking, $\hat{f}(x)$ is a prediction of $f(x)$, the true regression mean of y as shown in (1). A confidence interval (CI) is an estimate, obtained from observed data, of an interval where the true mean of $f(x)$ must lie with a given confidence. In this sense, the CI deals with the uncertainty of $\hat{f}(x)$ to estimate $f(x)$, i.e. the epistemic uncertainty. A prediction interval (PI), on the other hand, quantifies the uncertainty of using $\hat{f}(x)$ to predict y; it defines an interval where the observations of y must fall inside the interval with a given probability. Therefore, in addition to the epistemic uncertainty, the PI must also account for the aleatory uncertainty.

According to the concepts above, a CI is based on the characteristics of $P(f(x)|\hat{f}(x))$, and a PI is based on the characteristics of $P(y|\hat{f}(x))$. It is easy to see that PIs are wider than CIs. In this work the focus is on PIs as we are interested in the uncertainty of using $\hat{f}(x)$ for predicting y.

The reason for the focus on interval models, PIs in this case, is that they are a useful and practical tool for expressing the uncertainty of the predictions with the minimum information. Only three outputs are needed: an upper bound ($\bar{y}$), the lower bound ($\underline{y}$) and the coverage probability (PICP, as it will be defined later). This is enough to inform the most probable range and the probability of enclosure. In contrast, probability density functions (PDFs) contains exact information about the uncertainty. However, in practice, only particular distributions can be described with few parameters. For the general case, a large number of points is needed to approximate (up to a grid precision) the PDFs. Therefore, PIs are the most understandable uncertainty quantification mechanism that uses few parameters in the general case.

There is a wide variety of methods to construct PIs. For this, we want to find the lower and upper bounds $\underline{y}, \bar{y}$ so that $y \in [\underline{y}, \bar{y}]$ with probability $1 - p$, i.e.

$$P(\underline{y} \leq y \leq \bar{y}) = 1 - p. \tag{3}$$

A basic method to find these bounds relies on the use of the PDF of y, which is assumed known. If the probability that y is outside this interval is chosen to be the same above and below the interval, the bounds can be found as $\underline{y} = \mathcal{F}^{-1}(p/2)$ and $\bar{y} = \mathcal{F}^{-1}(1 - p/2)$, where $\mathcal{F}^{-1}(\cdot)$ is the inverse of the distribution function of y. This can be performed with known parametric PDFs or with PDFs with any non-parametric estimation. However, in practice a PDF cannot be estimated perfectly, and therefore there will be error propagation when computing the interval. For this reason, if one is only interested in the PI and does not need all the information in the PDF, it is more convenient to use a method that estimates the PI without needing a known PDF.

C. NOTATION FOR DYNAMICAL SYSTEMS

In this review we focus on methods based on computational intelligence that directly construct the PIs for dynamical systems, in particular those based on fuzzy models and neural networks. Here, in order to establish the notation to be used for the inputs variables of the different methods to be covered, it is assumed that the system to be modeled follows a dynamic described by a nonlinear autoregressive exogenous model (NARX), i.e. the output of the system is given by

$$y(k) = f(z_k), \tag{4}$$

where z_k is the input vector of size n_z, composed by the past n_y outputs y and the past n_u exogenous variables u, such that the input vector is defined as

$$\begin{aligned} z_k = \big[& y(k-1), y(k-2), \ldots, y(k-n_y), \\ & u(k-1), u(k-2), \ldots, u(k-n_u)\big]^T . \end{aligned} \tag{5}$$

D. PREDICTION INTERVAL METRICS

Several metrics have been used in the literature to design, validate and compare the effectiveness of prediction intervals. The most standard metrics, which will be used in this work, are presented next.

The prediction error of the crisp model is one of the most important factors that affects the behavior of the uncertainty band. In order to represent the accuracy of the predictive model, the root mean square error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y(k) - \hat{y}(z_k))^2}, \tag{6}$$

is selected in this work as the metric for representing the accuracy of the predictive model. In (6), N is the number of data selected for the computation of this metric, $y(k)$ is the real measurement of the system of signal output, and $\hat{y}(z_k))$ is the predicted value given by the identified model when the input vector z_k is received.

The interval width and coverage level are remaining most important metrics. In [4], [8], the Prediction Interval Normalized Averaged Width (PINAW) and the Prediction Interval Coverage Probability (PICP) are defined respectively for representing those factors, based on the following expressions:

$$\text{PINAW} = \frac{1}{NR}\sum_{k=1}^{N}\left(\bar{y}(z_k) - \underline{y}(z_k)\right), \tag{7}$$

where

$$R = \max\{y(k)\} - \min\{y(k)\} \tag{8}$$

is the range of values of the training data, and

$$\text{PICP} = \frac{1}{N}\sum_{k=1}^{N} c_k, \tag{9}$$

where

$$c_k = \begin{cases} 1, & \text{if } \underline{y}(z_k) \leq y_k \leq \bar{y}(z_k) \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

indicates if the interval given by the bounds $[\underline{y}(z_k), \bar{y}(z_k)]$, contains the real measure of $y(k)$.

E. CLASSIFICATION OF METHODS FOR THIS SURVEY

Given that there is a wide variety of methods for PI construction as discussed above, a proper classification of the methods to be included in this survey is in place. This classification will be based on two main criteria: (i) type of model used to describe the prediction interval, and (ii) the type of training procedure for PI construction.

Given the scope of this survey, the type of base model used for the PI methods defines the following division considered for this paper: (a) methods based on fuzzy models, and (b) methods based on neural networks.

On the other hand, considering the type of training, first categorization corresponds to Sequential methods, which receives a previously identified base crisp model that estimates $f(x)$ and then uses it to constructed the PI in a subsequent step. The second category, which will be referred to as Direct methods, perform the construction of the interval models in parallel with the crisp model identification. Therefore, a previously identified model is not required for this case. In this type of methods, the training process of the model incorporates the calculations to estimate the uncertainty band of the predictions, so that the models obtained are capable of directly outputting a prediction interval without the need of additional steps.

This classification scheme is shown in Table 1. According to it, the methods studied in the context of this work are presented in the following order: the sequential and direct methods developed based on fuzzy models are presented first in section III. Then, the sequential and direct methods based on neural networks are described in section IV.

TABLE 1. Classification of PI construction methods considered for this survey.

Methods based on Fuzzy Models		Methods based on Neural Networks	
Sequential Methods	Direct Methods	Sequential Methods	Direct Methods

III. FUZZY PREDICTION INTERVALS

A. BACKGROUND

Rule-based FLS are considered for solving a broad range of problems, from forecasting, to classification, control, etc. Because of this, the use of prediction intervals based on these models arises as an important way to include information about uncertainty in the implementation of forecasting systems and control strategies.

Among rule-based fuzzy models, the formulation presented by [9] arise as one of the most used structure for rule base fuzzy modeling. The fuzzy model of Takagi and Sugeno of this formulation is represented by the weighted sum of several linear models, as shown as follows

$$\text{Rule r: If } z_k \text{ is } A_r, \text{ then: } \hat{y}_r(z_k) = [1\ z_k^T]\theta_r, \tag{11}$$

$$\hat{y}(z_k) = \sum_{r=1}^{m} \beta_r(z_k)\hat{y}_r(z_k) \tag{12}$$

where z_k is the input vector of the fuzzy model at instant k (see equation (5)), A_r is a fuzzy set (antecedents) and $\hat{y}_r$ is the output of the local model (consequences) for rule r. The model consists of m rules so that $r = 1, \dots, m$. For each rule r in (11), θ_r is the parameter vector, which can be obtained using a suitable regression algorithm, and $\beta_r(z)$ corresponds to the degree of activation of the rule. The values of $\beta_r(z)$ are calculated from the membership functions (MF) of the fuzzy sets which are part of the antecedents of the rule r.

In order to consider the effect of uncertainties in the model identification, the type-2 fuzzy models presented in [10] arise as an appropriate extension of the fuzzy rule-based models. In this case, type-2 fuzzy sets, which are originally introduced in [11], are used for the antecedents of the rules. Thus, as shown in Figure 1, a type-2 fuzzy set A has a membership function defined by two different values, corresponding to its upper and lower limits ($\overline{\mu}_A(x)$ and $\underline{\mu}_A(x)$ respectively).

The value of the membership function lies inside this gray area bounded by $\underline{\mu}_A(x)$, $\overline{\mu}_A(x)$, which is denominated in literature as the *domain of uncertainty* of A (DOU(A)) or the *footprint of uncertainty* of A (FOU(A)). Due to the use of these type-2 fuzzy sets the local outputs of the rules are now defined by an upper and lower value. Note that, this kind of models are suited for the later implementation of the prediction intervals.

Fuzzy models have also been implemented in the literature as fuzzy neural networks (FNN) [12], [13]. These implementations result from the adaptation of TS fuzzy models to the artificial neural networks structure. The stages defined for TS models are handled by different layers of the neural network in FNNs, such that each of the antecedents and the consequences of the rules have their own neurons associated.

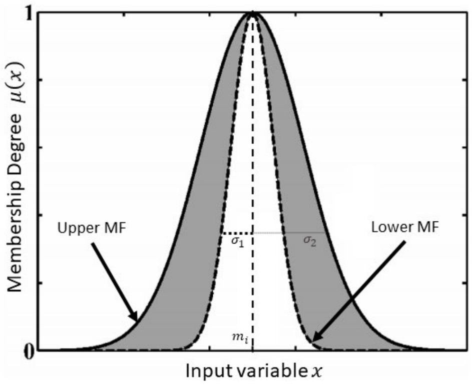


FIGURE 1. Type-2 membership function.

In the specialized literature, several methods based on Takagi-Sugeno fuzzy models have been proposed in order to obtain a Prediction Interval. As mentioned in section II-E, those methods are classified by the type of training procedure for its construction.

The fuzzy interval models of both categories, corresponding to sequential and direct methods, are reviewed below.

B. SEQUENTIAL METHODS

1) COVARIANCE METHOD

In [14], the upper and lower bounds that define the interval are constructed based on the error covariance of each rule of the Takagi-Sugeno fuzzy model. This strategy of fuzzy PI design is referred in the literature as the covariance method and it is based on the work done by Škrjanc in [15]. For an input vector z_k, the bounds of the local model r, for each measurement k are defined as

$$\overline{y}_r(z_k) = \hat{y}_r(z_k) + \alpha\mathcal{I}_r(z_k), \tag{13}$$

$$\underline{y}_r(z_k) = \hat{y}_r(z_k) - \alpha\mathcal{I}_r(z_k). \tag{14}$$

where α is a tuning parameter and $\mathcal{I}_r$ is an estimate of the variance of $\hat{y}_r$. The estimate $\mathcal{I}_r$ is given by

$$\mathcal{I}_r(z_k) = \hat{\sigma}_r\left(I + \psi_r^T(z_k)\left[\Psi_r\Psi_r^T\right]^{-1}\psi_r(z_k)\right)^{1/2}, \tag{15}$$

where I is an identity matrix of appropriate size; $\hat{\sigma}_r^2 I$ is the variance of the noise signal; and $\Psi_r = [\psi_r(z_1), \dots, \psi_r(z_M)]^T$ is the regression matrix that contains the regression vectors (past values of $\psi_r^T(z) = [1\ z_k^T]$ in the TS fuzzy model) for the M samples of a training dataset.

This method of fuzzy prediction interval design resembles the defuzzification stage performed in the aforementioned methods, which follow the idea presented in (12). Here, a global value for the estimated variance of $\hat{y}$ is obtained as

$$\mathcal{I}(z_k) = \sum_{r=1}^{m} \beta_r(z_k)\mathcal{I}_r(z_k). \tag{16}$$

Thus, the bounds of the global output of the fuzzy interval model is given by

$$\overline{y}(z_k) = \hat{y}(z_k) + \alpha\mathcal{I}(z_k), \tag{17}$$

$$\underline{y}(z_k) = \hat{y}(z_k) - \alpha\mathcal{I}(z_k). \tag{18}$$

This method for the prediction interval construction has the main advantage of a low computational cost for obtaining the interval. This is because the uncertainty is estimated directly by the given covariance formula. Thus, the main complexity of this method lies in the search of the proper values for the tuning parameters α. On the other hand, this interval structure presents a weakness in differentiating the effect of internal and external sources of uncertainty. This is a clear disadvantage of this method, because the performance of the prediction interval will be negatively affected when trying to model a system with exogenous input variables that have their own uncertainty.

This method has been applied in [14] for the forecasting of renewable generation and demand data from an installed microgrid and in [16] for the robust control of a solar collector field. Also, in [17] is applied in a fault detection system for an aircraft, and in [18] is used for an implementation of an indoor localization algorithm.

Additionally, some versions of this method have been developed in order to reduce the interval width. For example, in [19] different tuning parameters were proposed for the upper and lower bounds ($\overline{\alpha}$ and $\underline{\alpha}$ respectively), achieving a non-symmetrical interval around the prediction value. This version of the method has been applied in [19] for the identification of intervals for traffic measurements, and in [20], [21] for the forecasting of renewable generation and demand data from an installed microgrid. Later, a third version of this method was proposed in [22], where the tuning parameters can also vary depending of the instant k when the predictions are made (α_k). This is done in order to adjust the interval width to the specific cases of signals with uncertainty behaviors strongly dependent on the time of day, such as the cases of solar generation and occupancy level of offices. This version was only used in [22] for a robust predictive control design applied on a climatization system.

2) METHOD BASED ON INTERVAL FUZZY NUMBER FOR THE OUTPUT UNCERTAINTY

In [23], an interval is proposed from a Takagi-Sugeno fuzzy model. In that case, after that the model is identified, a single interval fuzzy number is added in order to approximate the model uncertainties present in the output of the system. That interval was later denoted in [24] as A1-C1 Takagi-Sugeno (TS), because is defined by type-1 antecedents and consequences. This interval is defined in [23] as

$$y_r = f_r(x) + a, \quad a \in A_0 = \left\{a_0, \in [\underline{a_0}, \overline{a_0}], \mu_A(t)\right\}, \tag{19}$$

where y_r is the output and f_r is the local model identified, both for each rule r. Also in (19), A_0 is a type-1 fuzzy set defined by the bounds $[\underline{a_0}, \overline{a_0}]$ (which are obtained from the knowledge of the uncertainties given by the error of the model) and the membership function $\mu_A(t)$.

This method presents its low computational cost to obtain the interval as its main advantage. This is because the construction of the interval is reduced to identifying a single fuzzy number. However, this feature also causes the main disadvantage of this method, which is that its use is very situational. A reliable prediction interval can be generated when external disturbances act as the only source of uncertainty affecting the system.

The intervals obtained with this method have been applied for the forecasting of solar power generation in [23], [24].

3) METHOD BASED ON FUZZY NUMBERS (SEQUENTIAL VERSION)

In the work [2], another version of fuzzy interval is proposed, based on the idea of intervals fuzzy numbers presented in [25]. In this method, the parameters of the consequences are defined by fuzzy numbers, thus, the output of the fuzzy interval for each rule r is given by the bounds

$$\overline{y}(z_k) = \sum_{i=1}^{n_z} g_r^{(i)} z_k^{(i)} + g_r^{(0)} + \sum_{i=1}^{n_z} \overline{s}_r^{(i)} |z_k^{(i)}| + \overline{s}_r^{(0)}, \tag{20}$$

$$\underline{y}(z_k) = \sum_{i=1}^{n_z} g_r^{(i)} z_k^{(i)} + g_r^{(0)} - \sum_{i=1}^{n_z} \underline{s}_r^{(i)} |z_k^{(i)}| - \underline{s}_r^{(0)}, \tag{21}$$

where z_k is the input vector of the fuzzy model at instant k. In (20)-(21), $g_r^{(i)}$ are the mean values and $\overline{s}_r^{(i)}$,$\underline{s}_r^{(i)}$ are the spread values of the parameter $\theta_r^{(i)}$ which is associated to the i-th component of z_k. Then, the global bounds of the fuzzy interval are obtained as

$$\overline{y}(z_k) = \sum_{r=1}^{m} \beta_r(z_k)\overline{y}_r(z_k), \tag{22}$$

$$\underline{y}(z_k) = \sum_{r=1}^{m} \beta_r(z_k)\underline{y}_r(z_k), \tag{23}$$

In the direct version of this method, the fuzzy model has already been identified, so the mean values of the parameters $g_r^{(i)}$ are assumed known. Thus, the interval identification consist in obtaining the spread values ($\overline{s}_r^{(i)}$, $\underline{s}_r^{(i)}$) from solving the following optimization problem:

$$\begin{aligned} &\min_{\overline{s}_r,\underline{s}_r} \text{PINAW} \\ &s.t. \ \text{PICP} = (1-\alpha)\%, \end{aligned} \tag{24}$$

The equality constraint PICP $= (1 - \alpha)\%$ is a hard constraint, and due to the nonlinear characteristic of the optimization problem (24), it could be difficult to solve with the typical algorithms. With the purpose of relaxing this equality constraint, a version of the optimization problem is proposed in [2], which includes that constraint as a barrier function. Such that, the interval identification can also be defined by the following optimization problem:

$$\min_{\overline{s}_i,\underline{s}_i} J = \eta_1 \text{PINAW} + exp\{-\eta_2\, [\text{PICP} - (1-\alpha)]\}. \tag{25}$$

This method for the prediction interval construction presents an increment of the computational cost when compared with the previous methods, due to the higher number of parameters that need to be identified. This inclusion of more parameters, like the separated spread values, allows to obtain more precise prediction intervals when dealing with systems with complex uncertainty behavior (such as non-Gaussian or heteroscedastic noise).

This interval method based on fuzzy numbers was applied in the case study presented in [2], which consists in the forecasting of the load in a microgrid and the energy consumption in some residential dwellings.

4) METHOD BASED ON α-LEVEL CUTS OF THE OUTPUT

In the work of [26] a Mamdani fuzzy logic system based on α-level cuts of the output is presented to generate the prediction intervals. An α-cut of a fuzzy set A is a crisp set A_α that contains all the elements in U that have membership values in A greater than or equal to α, that is [27],

$$A_\alpha = \{x \in U \mid \mu_A(x) \geq (\alpha)\}. \tag{26}$$

The general idea to obtain the prediction interval based on alpha-cut (α) approach is to decompose output fuzzy sets into a collection of crisp sets related together via the α levels [28]. Therefore, Alpha cuts on Mamdani Fuzzy logic system based output fuzzy sets (commonly defuzzified to a crisp number) are used to provide this prediction interval using the following 4-step process [26]:

- FLS Desing: A Mamdani FLS is designed to obtain the expected value of the system prediction. Standard approaches to FLS prediction are applied.
- Generating Output Fuzzy Sets: After constructing the standard Mamdani FLS, the inferencing over the input Fuzzy Sets and the rules are processed with conventional operators. By implementing the selected operators in the inference step, the output Fuzzy Sets of the Mamdani FLSs are generated.
- Defining alpha-cut level: There are several options to select an appropriate level of α for a given application. In general, the higher the selected value of α, the more narrow the output prediction interval. However, some α levels (those which are greater than the height of the output Fuzzy Set) may not result in any prediction interval output (alpha-cut is an empty set). In addition, some α levels may lead the traditional centroid defuzzification results falling outside of the generated prediction intervals.
- Generate alpha cut and centroid based outputs: Based on the selected α-level, the support is used as the prediction interval. Thereafter, a defuzzification technique is implemented on the gathered output set and a crisp value is calculated.

The proposed method was explored using both synthetic (Chaotic Mackey-Glass) and smart-grid specific real-world (wind power) time series datasets for one-step ahead.

With the sequential methods based on the fuzzy models already covered in this review, now the intervals of the second category aforementioned (the direct methods) will be explained below.

C. DIRECT METHODS

1) MIN-MAX METHOD

The main idea of this kind of methods was introduced in [29], where a min-max method is proposed for identifying the bounds of the fuzzy interval. Then, the interval is obtained by the identification of two different fuzzy functions, called the upper and lower functions ($\overline{f}(z)$ and $\underline{f}(z)$ respectively), which are found by solving the following min-max optimization problems:

$$\min_{\overline{f}} \max_{z_i \in Z} |y_i - \overline{f}(z_i)| \ \text{ subject to } \ y_i - \overline{f}(z_i) \leq 0, \quad \forall i, \tag{27}$$

$$\min_{\underline{f}} \max_{z_i \in Z} |y_i - \underline{f}(z_i)| \ \text{ subject to } \ y_i - \underline{f}(z_i) \geq 0, \quad \forall i, \tag{28}$$

where z_i is the i-th sample of the training dataset Z, and y_i is the i-th measured output. Thus, the bounds of the prediction interval associated to the input z_k are given by

$$\overline{y} = \overline{f}(z_k) = \sum_{r=1}^{m} \beta_r(z_k)\overline{\theta}_r z_k, \tag{29}$$

$$\underline{y} = \underline{f}(z_k) = \sum_{r=1}^{m} \beta_r(z_k)\underline{\theta}_r z_k, \tag{30}$$

where $\overline{\theta}_r$ and $\underline{\theta}_r$ are the parameter vectors of the fuzzy functions obtained in (27) and (28) respectively.

The advantage of this method is its great flexibility for modeling the behavior of prediction error variance. This feature is present thanks to the identification of two different models for representing the bounds of the prediction uncertainty. However, the use of two computationally min-max problems, increases the requirements that the selected optimization algorithm must meet in order to obtain accurate results. Furthermore, this method does not offer a direct mechanism to properly sets the expected values for the width and coverage level of the intervals. Thus, if a specific coverage level is wanted for the prediction interval, some variations of the (27)-(28) should be developed first.

The min-max method was applied to the fault detection problem over various type of systems. In [30] this method was used to detect the fault in a Motor-Generator Plant, in [31]–[33] the formulation of a fault-detection system for a nonlinear system with uncertain parameters, and in [34] this method was used for the construction of a belief rule-based model for the identification problem of uncertain nonlinear systems. Also, in [35]–[37] the fuzzy intervals based in this method were used in the implementation of a robust control strategy and in [36]–[38] were applied in the estimation of time-series related with renewable energy systems.

2) METHOD BASED ON INTERVAL-VALUED DATA

In [39] another version of fuzzy interval models was proposed, based in the use of interval arithmetic for the modeling of an interval valued output. Under this formulation, the output signal Y to be modeled must be an interval-valued data defined by a center and a radius, i.e. $Y = (Y_a, Y_c)$. In this approach, and following the formulation of TS fuzzy models (11)-(12), different local models are identified for the interval-valued output. Thus, for each rule r, the respective local model has the form

$$\hat{Y}_r(z_k) = \theta_r^T z_k, \tag{31}$$

where each component of the vector $\theta_r^T = [\theta_r^{(0)}, \dots, \theta_r^{(n)}]$ are interval-valued parameters defined by a corresponding center $a_r^{(i)}$ and a radius $c_r^{(i)}$, such that

$$\theta_r^{(i)} = (a_r^{(i)}, c_r^{(i)}). \tag{32}$$

Note that, this definition of the model is only valid when the input vector z_k is made up of crisp values. However, the dynamical interval Takagi-Sugeno fuzzy model proposed in [39] considers a mixed typed identification problem (there, z_k can be made up of interval-valued and crisp data). Considering this, the local rules presented in (31) can be rewritten as follows:

If $Y(k-1)$ is $A_{r,1}, \dots, Y(k-n_Y)$ is A_{r,n_Y}, $u(k-1)$ is $B_{r,1}, \dots, u(k-n_u)$ is B_{r,n_u}, then:

$$\hat{Y}_r(z_k) = P_{r,0} + \sum_{j=1}^{n_Y} Y(k-j) \circ P_{r,j} + \sum_{l=1}^{n_u} u(k-l) P_{r,n_Y+l}, \tag{33}$$

where $A_{r,i}$ and $B_{r,i}$ are the fuzzy sets associated to $Y(k-j)$ and $u(k-j)$ respectively. Note that, since $Y(k-j)$ is an interval-valued data, $A_{r,i}$ must be composed by two different fuzzy sets, such that, one is associated to the center value of $Y(k-j)$ (denoted as $A_{r,i}^a$), meanwhile the other correspond to its radius (denoted as $A_{r,i}^c$). Also in (33), the values $P_{r,i}$ for $i = 1, \dots, n_Y$ are the crisp components of the parameter vector θ_r which multiply the interval-valued data $Y(k-i)$. On the other hand, the values P_{r,n_Y+l} for $l = 1, \dots, n_u$ are the interval-valued parameters that multiply the crisp input variable $u(k-l)$.

In order to obtain the prediction interval, the operator $\circ$ included in (33) is proposed from the interval arithmetic developed in [39]. That operator is defined for an interval-valued data (a, c) and a vector $x^T = [x_{11}, x_{12}, x_{21}, x_{22}, \dots, x_{n1}, x_{n2}]$ as follows

$$(a, c) \circ x = \begin{bmatrix} (ax_{11}, c|x_{12}|) \\ \vdots \\ (ax_{n1}, c|x_{n2}|) \end{bmatrix}. \tag{34}$$

Following the formulation of the global output of the Takagi-Sugeno fuzzy model presented in (12), the output of the proposed interval is represented by

$$\hat{Y}(z_k) = \sum_{r=1}^{m} \beta_r \hat{Y}_r(z_k) \tag{35}$$

This interval method has the low computational cost for the training process as its main advantage. This low cost is presented because the method uses data which are already been characterized by an interval, so the identified model gives directly the bounds for the predictions thanks to the interval arithmetic. Due to that, this interval construction method is similar in complexity to the identification process of a classic fuzzy TS model. However, this interval structure has the difference of a large quantity of parameters that need to be identified. On the other hand, because this method requires the use of interval-valued data, its applicability is restricted to the availability of these types of signals.

This interval method presented in (35) is applied in [39] for the identification of a nonlinear system. A similar strategy was followed in the works of [40], [41], where a fuzzy model is identified for an interval-valued data characterized by confidence intervals obtained from an electro-mechanical throttle valve using the Chebyshev's inequality.

3) METHOD BASED ON FUZZY NUMBERS (DIRECT VERSION)

In the work [42], an alternative version of the method based on intervals fuzzy numbers is proposed. The parameters of the consequences are defined by fuzzy numbers, thus, the output of the fuzzy interval for each rule r is given by the bounds

$$\overline{y}_r(z_k) = \sum_{i=1}^{n_z} g_r^{(i)} z_k^{(i)} + g_r^{(0)} + \sum_{i=1}^{n_z} s_r^{(i)} |z_k^{(i)}| + s_r^{(0)}, \tag{36}$$

$$\underline{y}_r(z_k) = \sum_{i=1}^{n_z} g_r^{(i)} z_k^{(i)} + g_r^{(0)} - \sum_{i=1}^{n_z} s_r^{(i)} |z_k^{(i)}| - s_r^{(0)}, \tag{37}$$

where $g_r^{(i)}$ are the mean values and $s_r^{(i)}$ are the spread values of the parameter $\theta_r^{(i)}$. Then, the global bounds of the fuzzy interval are given by the same bounds (22)-(23) used in the sequential version of this method.

In this version of the method, the construction of the interval fuzzy model, considers the identification of both values $g_r^{(i)}$ and $s_r^{(i)}$ performed at the same time (without the classical identification of the TS model required in the sequential version). In order to achieve that, an optimization problem is proposed in [42], which minimize the following multi-objective cost function,

$$V = \eta_1 \, \mathrm{MSE} + \eta_2 \, \mathrm{PINAW} + \eta_3 \, (\nu(\mathrm{PICP} - \mathrm{PICP_D}))^2, \tag{38}$$

where $\mathrm{PICP_D}$ is the desired coverage probability for the prediction interval, ν regulates the size of the allowed PICP error and η_1, η_2, and η_3 are weighing factors. Because the optimization is nonlinear, the problem is solved in [42] using the Particle Swarm Optimization (PSO) and the Improved Teaching Learning Based Optimization.

This method shares the potential to achieve the same benefits as its sequential version. The main difference of this version is its complexity for the training process. This occurs because the method performs the obtaining of the model and the prediction interval at the same time, under the same procedure.

This interval based on fuzzy numbers was applied only in the case study presented in [42], where a load forecasting in a microgrid is performed.

4) METHOD BASED ON TYPE-2 TAKAGI & SUGENO MODELS

In the specialized literature, prediction intervals have been implemented based on type-2 Takagi & Sugeno models. According to [43], the interval type-2 (IT2) fuzzy systems can be implemented by using type-2 fuzzy sets as antecedents of the rules. Due to that, two activation degrees can be obtained for each rule, $\underline{\beta}_r(z_k)$ and $\overline{\beta}_r(z_k)$, which work as the lower and upper limits of the membership functions. Also, the consequences of the rules are defined by the parameter vector θ_r that corresponds to an interval type-1 fuzzy number, i.e. each component is defined by an upper and lower bounds ($\overline{\theta}^{(i)}$ and $\underline{\theta}^{(i)}$ respectively). Considering that, the limits for each rule can be computed as follows:

$$\overline{y}_r(z_k) = \overline{\theta}_r^{(0)} + \overline{\theta}_r^{(1)} z_k^{(1)} + \ldots + \overline{\theta}_r^{(n)} z_k^{(n_z)}, \tag{39}$$

$$\underline{y}_r(z_k) = \underline{\theta}_r^{(0)} + \underline{\theta}_r^{(1)} z_k^{(1)} + \ldots + \underline{\theta}_r^{(n)} z_k^{(n_z)}. \tag{40}$$

Based on the limits identified for each rule, there are two ways to compute the global bounds of the IT2 fuzzy model. The unnormalized way considers the following global bounds for the model

$$\underline{y}(z_k) = \sum_{r=1}^{m} \underline{\beta}_r(z_k)\underline{y}_r(z_k), \tag{41}$$

$$\overline{y}(z_k) = \sum_{r=1}^{m} \overline{\beta}_r(z_k)\overline{y}_r(z_k), \tag{42}$$

while the normalized version computes the global limits as follows

$$\overline{y}(z_k) = \frac{\sum_{r=1}^{U} \underline{\beta}_r(z_k) y_r(z_k) + \sum_{r=U+1}^{m} \overline{\beta}_r(z_k) y_r(z_k)}{\sum_{r=1}^{L} \underline{\beta}_r(z_k) + \sum_{r=L+1}^{m} \overline{\beta}_r(z_k)}, \tag{43}$$

$$\underline{y}(z_k) = \frac{\sum_{r=1}^{L} \overline{\beta}_r(z_k) y_r(z_k) + \sum_{r=L+1}^{m} \underline{\beta}_r(z_k) y_r(z_k)}{\sum_{r=1}^{L} \overline{\beta}_r(z_k) + \sum_{r=L+1}^{m} \underline{\beta}_r(z_k)}, \tag{44}$$

where L, U are the switch points given by the Karnik-Mendel Algorithm [44]. Under this formulation, the prediction interval can be finally obtained by solving an optimization problem, similar as performed in (24), where the parameters of the membership functions $\underline{\beta}_r(z_k)$, $\overline{\beta}_r(z_k)$, and the consequences $\underline{\theta}_r$, $\overline{\theta}_r$, are the optimization variables.

This prediction interval method arises as the most complete methodology for characterizing the uncertainty of the predictions. This is because this interval structure allows to differentiate between the uncertainty present in the input variables (by the implementation of type-2 fuzzy sets for the antecedents) and the error associated with the accuracy of the identified model (by the use of interval type-1 fuzzy parameters). However, this feature also provokes the high computational cost for training of this method, due to the large number of parameters that must be identified, considering that each of them considerably affects the functioning of the model.

The IT2 TS fuzzy models have been widely used in literature for the forecasting of different types of signals (see as example [45]–[49], [49]). However, the scope of this review focuses on the application of this fuzzy models as prediction intervals.

The IT2 TS fuzzy models have been mainly applied as prediction intervals in several cases related with the modeling of renewable energy systems. For example, in [23], [24] these intervals were used for the modeling of solar power, in [50]–[53] were used for wind power forecasting, and also in [52] were used to characterize the loads in a microgrid.

5) METHOD BASED ON FUZZY NEURAL NETWORKS

Despite the use of type-2 fuzzy sets in the formulation of IT2 TS fuzzy models, its use in the literature is mainly focused on the construction of prediction intervals based on fuzzy neural networks.

Following the main ideas of IT2 TS models, as previously mentioned, the fuzzy neural networks can be implemented by the use of type-2 fuzzy sets in the membership layer (this is analog to the use of type-2 fuzzy sets as antecedents in the IT2 TS model). Also, the rule layer of the fuzzy neural network can be defined by type-1 fuzzy numbers (similar idea to the use of type-1 consequences in the IT2 TS models). Finally, the bounds of this prediction interval are then given by the output values of the neurons in the new Type-reduction layer. The structure resulting from the combination of the fuzzy neural network with the concepts of type-2 fuzzy models, is shown on Figure 2.

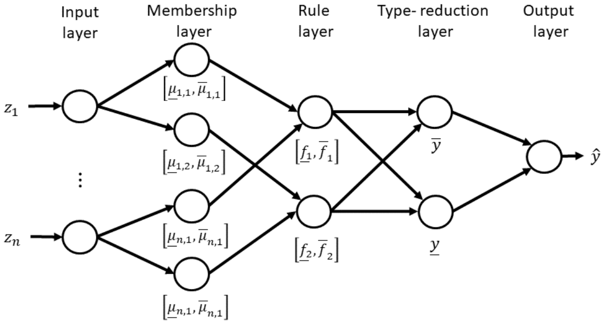

Input layer
Membership layer
Rule layer
Type- reduction layer
Output layer
z_1
z_n
$[\underline{\mu}_{1,1}, \overline{\mu}_{1,1}]$
$[\underline{\mu}_{1,2}, \overline{\mu}_{1,2}]$
$[\underline{\mu}_{n,1}, \overline{\mu}_{n,1}]$
$[\underline{\mu}_{n,1}, \overline{\mu}_{n,1}]$
$[\underline{f}_1, \overline{f}_1]$
$[\underline{f}_2, \overline{f}_2]$
$\overline{y}$
$\underline{y}$
$\hat{y}$

FIGURE 2. Interval Type-2 fuzzy neural network example.

This structure is denoted in the specialized literature as the Interval Type-2 Fuzzy Neural Network (IT2FNN), and has been applied in several cases of modeling systems. For example, the IT2FNN were used for the modeling of chaotic time series and nonlinear systems in [54]–[59]. Another specific applications include the uses for the modeling of an antilock braking systems in [60] and for wind power forecasting in [61]. Additionally, the work [62] presents an overview of this kind of prediction intervals used for the prediction of chaotic time series.

Note that, this interval method could be considered as an extension of the Type-2 Takagi Sugeno fuzzy models, because its implementation follows almost the same identification process. Each element that conforms the type-2 fuzzy models is now a different neuron, so the construction of this model lies in the task of obtaining the accurate neurons in order to achieve the expected response. This can be performed by a similar optimization as done with the previous type-2 fuzzy models. Because of that, this interval method shares the same characteristics with the original type-2 fuzzy models.

6) JOINT SUPERVISION METHOD

The Joint Supervision Method is based on two conflicting goals: Predictions must both be as close as possible to the expected (crisp) value, and the PI must be wide enough to comply with imposed coverage probability restrictions. The version of this method that uses fuzzy models is proposed in [63], where an adaptation of the joint supervision method, previously developed for neural networks in [64], is performed.

In [63] this method uses a fuzzy model with 3 outputs, where two of them are associated to the upper and lower interval bounds, while the third output corresponds to the expected target value.

For the training procedure, a different cost function is used for each output ($\overline{y}$, $\hat{y}$, $\underline{y}$). First, all model outputs share an MSE cost component to minimize prediction error. On the other hand, the outputs associated to the interval bounds introduce an additional MSE cost that is applied only when data points fall outside the predicted interval

$$\overline{L} = \frac{1}{N}\sum_{k=1}^{N}(y_k - \overline{y}(z_k))^2 + \lambda\frac{1}{N}\sum_{k=1}^{N} ReLU^2(y_k - \overline{y}(z_k)) \tag{45}$$

$$\hat{L} = \frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}(z_k))^2 \tag{46}$$

$$\underline{L} = \frac{1}{N}\sum_{k=1}^{N}(y_k - \underline{y}(z_k))^2 + \lambda\frac{1}{N}\sum_{k=1}^{N} ReLU^2(\underline{y}(z_k) - y_k) \tag{47}$$

where

$$ReLU(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases} \tag{48}$$

and λ is a hyperparameter that must be tuned for each individual experiment through an iterative process. On each iteration of that tuning process, a new Joint Supervision model is trained with a given λ value, which continues evolving until the model complies with the desired interval coverage level.

The main advantage of the Joint Supervision Method is the simplicity of its cost function, when compared with the optimizations performed during the training process of others direct methods based on fuzzy models. However, this training process still has a high computational cost due to the need to identify a new model for each value of λ tested. This version of the method is only applied in [63] for temperature forecasting. There, it is attempted to reduce the effect of the high computational associated to the training procedure by performing a logarithmic search for the hyperparameter λ.

With the prediction intervals based on fuzzy model already covered, next section continues with the presentation of the methods that use neural networks as its model base.

IV. NEURAL PREDICTION INTERVALS

A. BACKGROUND

Neural Network (NN) predictive models are artificial intelligence-based regression techniques that have seen continuous research and improvement since their original conception. Currently, neural network architectures can be divided between traditional models, which are based on small and shallow networks, and modern, Deep Learning-based models, which have caused a strong surge in research interest for these algorithms.

Among traditional architectures, Multi-Layer Perceptron (MLP) predictive models stand out as the most heavily researched alternative. They consist of an input layer, followed by a small (usually one) number of hidden layers, and a final output layer with a linear activation function. Under this formulation and assuming only one hidden layer is used, total model output can be expressed as follows

$$\hat{y}(z_k) = \sum_{j=1}^{N_h} w_j \left(f\left(\sum_{i=1}^{n_z} w_{ji} z_k^{(i)} + b_j \right) \right) + b \tag{49}$$

where z_k is the input vector of the neural network at instant k (see equation (5)), N_h is the number of neurons in the hidden layer, w_j is the weight that connects the j^{th} hidden neuron with the output, w_{ji} is the weight that connects the i^{th} input neuron with the j^{th} hidden neuron, n_y and n_u represent the input vector length according to the notation presented in equation (5), $z_k^{(i)}$ is the i^{th} component of input vector z_k, b_j is the bias parameter associated with the j^{th} hidden neuron, b is the bias associated with the output neuron, and $f(\cdot)$ is a nonlinear activation function, which is usually set as a hyperbolic tangent or the sigmoid function $S(x) = \frac{1}{1+e^{-x}}$. Currently, on the field of NN-based prediction interval modeling, a great majority of the existing literature is based on this architecture [8], [64]–[67].

On the other hand, modern architectures have focused on the research of different, more complex networks that are more suited for predictive modeling, such as recurrent networks [68], which are known for being capable of naturally modeling data with temporal dynamic behavior, and deep neural architectures, which have achieved better performance than traditional networks by allowing the use of denser models. Among these architectures, both shallow and deep Long Short-Term Memory (LSTM) networks [69] stand out as a popular alternative in predictive modeling that is also starting to see applications in prediction interval modeling [7], [70].

Aside from their structure, neural models can also vary on their training procedure. Since most of the modern work in literature is based on Deep Learning architectures, which require fast and efficient training algorithms due to the sheer amount of parameters they have, the most commonly used technique consists in stochastic gradient descent optimization with the help of the Backpropagation algorithm. This means the cost function's gradient must be constantly calculated, which demands the use of simple and differentiable functions, such as Mean Square Error (MSE).

However, while the previous statement is true in the field of neural predictive modeling, NN-based prediction interval models may require a more flexible approach. This is because interval models rely on modified network architectures adapted to produce a more informative output in the shape of interval upper and lower bounds, resulting in an increase in model complexity. Because of this, some methods must introduce more complex, nonlinear, or even non-differentiable cost functions whose derivative cannot be calculated in a closed form. Therefore, even though the stochastic gradient descent approach has proven to obtain good results while minimizing computational costs, research on neural interval models has also focused on evolutionary computing solutions, such as Genetic Algorithms [65] and Particle Swarm Optimization [66], [67]. Additionally, some niche cases also exist where the prediction interval construction method does not allow for any of these solutions, and instead requires a specific training methodology, such as Bayesian approaches [71], which rely on Markov Chain Monte Carlo sampling algorithms.

The following subsections show a review of the neural interval models that can be found in the specialized literature. As mentioned in section II-E, these methods have been categorized into sequential and direct methods.

B. SEQUENTIAL METHODS

1) DELTA METHOD

This method [72] uses a strategy consisting of using classical nonlinear regression theory to obtain an approximation of the prediction uncertainty. This can be done through the calculation of the Taylor series expansion of the mean square error network cost function. After this, assuming the noise variance is homogeneous and normally distributed, total prediction variance for each prediction can be represented as

$$\sigma_{tot}^2(z_k) = \sigma_\epsilon^2(1 + g_0^T(J^T(z_k)J(z_k))^{-1}g_0) \tag{50}$$

where g_0 represents the NN model parameters (weights and biases), $J(z_k)$ is the network's Jacobian matrix evaluated on input sample z_k, and σ_ϵ is the data noise variance, which can be obtained from

$$\sigma_\epsilon^2 = \frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}(z_k))^2 \tag{51}$$

where y_k represents the training data point for sample k, and $\hat{y}(z_k)$ is the model prediction for input sample z_k.

After calculating this value, the $(1-\alpha)\%$ confidence prediction interval can be built, assuming it is symmetrical and Gaussian, as

$$\overline{y}(z_k) = \hat{y}(z_k) + t_{n-p}^{1-\frac{\alpha}{2}}\sigma_{tot}(z_k) \tag{52}$$

$$\underline{y}(z_k) = \hat{y}(z_k) - t_{n-p}^{1-\frac{\alpha}{2}}\sigma_{tot}(z_k) \tag{53}$$

where n is the number of input samples that were used to train the model, p is the number of network parameters, and $t_{n-p}^{1-\frac{\alpha}{2}}$ represents the $\frac{\alpha}{2}$ quantile of a Student's t-distribution cummulative distribution function with $n-p$ degrees of freedom.

Due to relying on the assumption that the uncertainty must be homogeneous and Gaussian, this method can fail to produce quality PIs for applications where data noise varies through time. Additionally, since the Delta Method requires the calculation of the model Jacobian matrix, PI calculation can be computationally expensive, which can make this method unfeasible for some online applications.

This method has seen applications in electricity load forecasting [72], solder paste deposition process monitoring [73], indoor temperature and relative humidity prediction [74], groundwater quality monitoring [75] and travel time prediction [76].

Finally, some versions of this method have been reported in literature that improve its performance. For example, in [77], the authors propose the inclusion of a weight decay term into the network cost function to prevent overfitting, while in [78] Khosravi *et al.* propose training a Delta method interval model where PICP and PINAW (shown in equations (7)-(10)) are used for neural network architecture optimization and in [79] a new cost function is proposed, which strives to minimize PI width while trying to keep parameters as close as possible to the values that do not violate the Delta Method's assumptions. However, despite these improvements, the main limitations of this method still remain.

2) BAYESIAN METHOD

The Bayesian Method involves treating the Neural Network model as a Bayesian Estimator [71], [80], [81], which means that the neural network weights are modeled as a probability distribution instead of single values, so that prediction uncertainty, and consequently PIs, can be intuitively derived from this distribution.

To optimize model parameters, the Bayesian Method attempts to estimate a posterior distribution for the network weights w given the training data points z_k by applying Bayes' Theorem. Using this strategy, it is possible to obtain an expression that is proportional to the desired posterior distribution, after which a Markov Chain Monte Carlo algorithm can be used to estimate the real posterior via a sampling procedure [71].

Once the approximate posterior is determined, total error variance can be easily calculated as

$$\sigma^2_{tot}(z_k) = \sigma^2_D + \sigma^2_{w^{MP}}(z_k) \tag{54}$$

$$= \frac{1}{\beta} + \nabla\hat{y}^T_{w^{MP}}(z_k)(H^{MP})^{-1}\nabla\hat{y}_{w^{MP}}(z_k) \tag{55}$$

where σ^2_D is the data noise variance, w^{MP} represents the Maximum a Posteriori network weights, $\nabla\hat{y}_{w^{MP}}(z_k)$ is the network gradient with respect to parameters w^{MP} evaluated on input sample z_k, and H^{MP} is the model Hessian matrix.

Finally, by specifying a desired coverage probability, a PI can be constructed by calculating the corresponding quantiles from the posterior distribution [4], [71]

$$\overline{y}(z_k) = \hat{y}(z_k) + q^{1-\frac{\alpha}{2}}\sigma_{tot}(z_k) \tag{56}$$

$$\underline{y}(z_k) = \hat{y}(z_k) - q^{1-\frac{\alpha}{2}}\sigma_{tot}(z_k) \tag{57}$$

where $q^{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ quantile of the normalized posterior distribution.

Despite having reported a performance better than that of the traditional Delta Method and having an understandable mathematical foundation based on Bayesian statistics, the calculation of the Hessian Matrix makes the Bayesian Method computationally demanding, which hinders its applicability to real problems. This in turn translates into a small number of applications in literature, mainly electricity load forecasting [82], potential evapotranspiration forecasting [83] and bus travel time prediction [76].

3) MEAN-VARIANCE ESTIMATION METHOD (MVEM)

The MVEM [84] proposes a PI can be built if two NNs are trained sequentially: One for estimating point predictions (NN_y), and one for estimating prediction error variance (NN_σ).

The methodology used to train this model is to first train network NN_y with a mean square error loss like any standard neural model. Then, network NN_σ is trained to estimate error variance σ^2_k between each prediction $\hat{y}(z_k)$ and target y_k data pairs. This is achieved by considering a log-likelihood cost function

$$C(y,\hat{y}) = \frac{1}{2}\sum_{k=1}^{N}\left(\log(\hat{\sigma}^2(z_k)) + \frac{(y_k - \hat{y}(z_k))^2}{\hat{\sigma}^2(z_k)}\right) \tag{58}$$

Finally, after obtaining predictors $\hat{\sigma}^2(z_k)$ and $\hat{y}(z_k)$, a PI can be built the same way as the Bayesian Method:

$$\overline{y}(z_k) = \hat{y}(z_k) + q^{1-\frac{\alpha}{2}}\hat{\sigma}(z_k) \tag{59}$$

$$\underline{y}(z_k) = \hat{y}(z_k) - q^{1-\frac{\alpha}{2}}\hat{\sigma}(z_k) \tag{60}$$

Since the MVEM estimates prediction uncertainty from the output of a Neural Network, it is less computationally demanding than the Delta and Bayesian methods, while also being capable of adapting to non-homogeneous noises thanks to neural networks being universal approximators. Nevertheless, its main drawback is that MVEM relies on the assumption that model NN_y accurately estimates the true ground truth data points y_k. This is because the log-likelihood cost proposed in equation (58) assumes prediction error is normally distributed with mean $\hat{y}(z_k)$, which means that model NN_y is being treated as an unbiased estimator of the real expected value of data points y_k.

This method has seen applications in wind power forecasting [85], exchange rate forecasting [86], remaining useful life prediction [87], stock market index forecasting and landslide displacement prediction [88].

Finally, some versions of this method have been reported in literature that improve its performance. First, in [85], a new technique was proposed to train the prediction error variance network NN_σ by fine-tuning the weights using a Coverage Width-based Criterion (CWC) cost, which can be calculated as

$$\mathrm{CWC} = \mathrm{PINAW}\left\{1 + \gamma(\mathrm{PICP})\, exp\left(\eta(\mu - \mathrm{PICP})\right)\right\}, \tag{61}$$

where $\gamma(\mathrm{PICP})$ is represented as

$$\gamma = \begin{cases} 0 & \mathrm{PICP} \geq \mu, \\ 1 & \mathrm{PICP} < \mu. \end{cases} \tag{62}$$

Here, η and μ are two hyperparameters that control the behavior of the metric CWC. It is important to note that μ represents the nominal coverage level of the interval, while η establishes the weight of the penalization for the situations where the PICP does not reach its desired values.

Another version is proposed in [88] which uses an Echo State Network, a type of recurrent neural network, as a baseline model. Since recurrent architectures are a dynamic modelling technique, this variation reported better variance estimations on time series prediction tasks.

4) BOOTSTRAP METHOD

The bootstrap method [5] combines an ensemble Neural Network model with strategic sampling techniques in order to estimate both model predictions and prediction uncertainty with more precision than standard methods based on single-model predictions [89].

In the field of neural modeling, ensemble models consist in a series of methodologies that allow the usage of many neural networks to jointly solve a problem [90]. Among the different types of existing ensemble models, the bootstrap technique stands out due to its ability to quantify the ensemble's prediction uncertainty [89]. Even though several bootstrap versions exist to accomplish this goal, the basic algorithm can be described as follows:

1) Sample total data into S training subsets through a uniform sampling with replacement method.
2) Train one NN model for each of the resampled subsets (for a total of S models)

3) Once each model has been trained, the total output of the ensemble can be calculated as

$$\hat{y}(z_k) = \frac{1}{S-1}\sum_{s=1}^{S}\hat{y}_s(z_k) \tag{63}$$

where $\hat{y}(z_k)$ is the total ensemble prediction for input sample z_k and $\hat{y}_s(z_k)$ is the prediction of the s^{th} NN model for input sample z_k

Once the ensemble has been trained, the bootstrap method proposes that epistemic uncertainty $\sigma_{\hat{y}}^2$ can be estimated from the variance in the outputs of the individual networks as

$$\hat{\sigma}_{\hat{y}}^2(z_k) = \frac{1}{S}\sum_{s=1}^{S}(\hat{y}_s(z_k) - \hat{y}(z_k))^2 \tag{64}$$

After this, aleatory uncertainty is estimated by training a new network $\hat{\sigma}_D^2(z_k)$ to fit an estimator of the remaining residuals

$$\hat{r}^2(z_k) = \max\left(\left[y_k - \hat{y}_{validation}(z_k)\right]^2 - \hat{\sigma}_{\hat{y}}^2(z_k), 0\right) \tag{65}$$

where y_k is the training data point for sample k and

$$\hat{y}_{validation}(z_k) = \frac{\sum_{s=1}^{S} q_s(z_k)\hat{y}_s(z_k)}{\sum_{s=1}^{S} q_s(z_k)} \tag{66}$$

$$q_s(z_k) = \begin{cases} 1, & z_k \text{ is in the validation set of model } s \\ 0, & \text{otherwise} \end{cases} \tag{67}$$

This final training step is achieved by using a negative loglikelihood cost function similar to equation (58):

$$C(\hat{r}^2) = \frac{1}{2}\sum_{k=1}^{N}\left(\log(\hat{\sigma}_D^2(z_k)) + \frac{\hat{r}^2(z_k)}{\hat{\sigma}_D^2(z_k)}\right) \tag{68}$$

Finally, once both uncertainties have been estimated, a PI can be constructed in a similar fashion to other sequential methods

$$\hat{\sigma}_{tot}^2(z_k) = \sigma_D^2 + \sigma_{\hat{y}}^2 \tag{69}$$

$$\bar{y}(z_k) = \hat{y}(z_k) + t_S^{1-\frac{\alpha}{2}}\hat{\sigma}(z_k) \tag{70}$$

$$\underline{y}(z_k) = \hat{y}(z_k) - t_S^{1-\frac{\alpha}{2}}\hat{\sigma}(z_k) \tag{71}$$

where t_S represents the $\frac{\alpha}{2}$ quantile of a Student's t-distribution CDF with S degrees of freedom.

Since the Bootstrap Method requires several models to be trained, it is more computationally expensive than MVEM, although this has minimal impact on the evaluation phase, which can make Bootstrap a feasible solution for online problems. Additionally, since this method can potentially estimate more precise predictions, results can be better than traditional MVEM, although this hypothesis relies on that each of the S NN models in the ensemble have been trained with sufficient data. Finally, it is important to note that due to the way it is trained, the Bootstrap Method is the only neural interval method capable of separating epistemic and aleatory uncertainty components.

This method has been applied in many different fields, such as estimation of mean wave overtopping discharge for coastal structures [91], wind power forecasting [51], [92], lake level forecasting [93], bus travel time prediction [94], nuclear transient feedwater flow prediction [95], degradation estimation of components subject to fatigue load [96], electricity price forecasting [97], nickel-based superalloy design [98], aerosol optical depth retrieval [99]

Some versions have also been reported for this method. First, in [100] a more accurate estimate for epistemic uncertainty is proposed by dividing the ensemble into M smaller ensembles, so that a set of M predictions is obtained for each input sample z_k

$$\xi = \{\hat{y}_{ens}^{i}(z_k)\}_{i=1}^{M} \tag{72}$$

from which a set of P bootstrap re-samples can be obtained as

$$\Xi = \{\xi_j^*\}_{j=1}^{P} \tag{73}$$

$$\xi_j^* = \{\hat{y}_{ens}^{j_1}(z_k), \ldots, \hat{y}_{ens}^{j_M}(z_k)\} \tag{74}$$

so that epistemic uncertainty can be finally calculated as

$$\sigma_{\hat{y}}^2(z_k) = \frac{1}{P}\sum_{j=1}^{P}\sigma_j^{*^2}(z_k) \tag{75}$$

where

$$\sigma_j^{2^*}(z_k) = \frac{1}{M}\sum_{i=1}^{M}\left(\hat{y}_{ens}^{j_i}(z_k) - \hat{y}^j(z_k)\right)^2 \tag{76}$$

$$\hat{y}^j(z_k) = \frac{1}{M}\sum_{i=1}^{M}\hat{y}_{ens}^{j_i}(z_k) \tag{77}$$

On the other hand, in [101] a different method for the estimation of aleatory uncertainty σ_D^2 is proposed by training a neural network using the CWC cost function shown in equation (61) using a Simulated Annealing algorithm.

5) COVARIANCE METHOD

The Covariance Method [14], [102] is very similar to the Delta Method but takes a more statistical approach based on the work done by Škrjanc and Sáez *et al.* on fuzzy interval models [14], [15].

For this method to work, first a standard 3-Layer Perceptron model must be trained, so that predictions can be estimated as shown in equation (49)

Using this notation and considering a hyperbolic tangent activation function, hidden neuron output vector Z_k and model parameter vector θ can be defined as

$$Z_k = [Z_{1k}, \ldots, Z_{N_h k}, 1] \tag{78}$$

$$\theta = [w_1, \ldots, w_{N_h}, b] \tag{79}$$

where

$$Z_{jk} = \tanh\left(\sum_{i=1}^{n_z} w_{ji} z_k^{(i)} + b_j\right) \tag{80}$$

$$\hat{y}(z_k) = Z_k^T \theta \tag{81}$$

After this, assuming prediction error is homogeneous and normally distributed, an expression can be calculated for prediction error variance

$$\sigma_k = \sigma_e (1 + Z_k^T (Z^* Z^{*T})^{-1} Z_k)^{\frac{1}{2}} \tag{82}$$

where σ_e represents the data noise variance, Z_k^T is calculated using validation input data, and Z^* is the matrix of all hidden neuron outputs Z_k for each sample of the training set.

Finally, a PI can be built as

$$\overline{y}(z_k) = \hat{y}(z_k) + \alpha\sigma_k \tag{83}$$

$$\underline{y}(z_k) = \hat{y}(z_k) - \alpha\sigma_k \tag{84}$$

where α is a tunable parameter that needs to be adjusted through an iterative process to accommodate for desired PI coverage probability.

As it can be noted in equations (50) and (82), the Covariance Method has a similar uncertainty quantification approach to the Delta Method, where the difference lies in that the Delta Method proposes a parameter-driven methodology to calculate prediction error variance, while the Covariance Method uses a data-driven estimation. This similarities mean that both methods also share the same limitations, which is the reliance on the assumption that prediction error is homogeneous and Gaussian. Additionally, it is important to note that the incorporation of an iterative process into the training procedure makes this method more computationally expensive than the Delta Method.

This method has seen applications on electricity load forecasting [64].

6) FUZZY NUMBERS METHOD

The Fuzzy Numbers Method [2], similar as done in the fuzzy version showcased in section III-B3, proposes that a PI can be constructed by using a modified Neural Network where the network weights are modeled as interval fuzzy numbers. This means that, using the notation presented in (49) and (80), output weights w_j are now represented by its mean ($g^{(j)}$) and two spread ($\underline{s}^{(j)}$, $\overline{s}^{(j)}$) values.

In order to train a model using this method, two identification routines must be executed: The first training procedure is responsible for identifying mean ($g^{(j)}$) parameters, and consists of a traditional setup for NN regression consisting of least means square optimization through Stochastic Gradient Descent via the Backpropagation algorithm. The second procedure consists on training the network to identify spread ($\underline{s}^{(j)}$, $\overline{s}^{(j)}$) parameters, which are obtained by solving an unconstrained optimization problem shown in equation (25) where the objective function resembles the form of the CWC metric presented in (61).

Then, the upper and lower bounds for the PI model $\overline{y}$, $\underline{y}$, are calculated using similar equations to those used in the version of this method based on fuzzy models. Thus, for this case (20)-(21) are rewritten as

$$\overline{y}(z_k) = \sum_{j=1}^{N_h} g^{(j)} Z_k^{(j)} + g^{(0)} + \sum_{j=1}^{N_h} \overline{s}^{(j)} |Z_k^{(j)}| + \overline{s}^{(0)} \tag{85}$$

$$\underline{y}(z_k) = \sum_{j=1}^{N_h} g^{(j)} Z_k^{(j)} + g^{(0)} - \sum_{j=1}^{N_h} \underline{s}^{(j)} |Z_k^{(j)}| + \underline{s}^{(0)} \tag{86}$$

where $Z_k^{(j)}$ are the same outputs of the hidden layer presented in (80).

Due to the introduction of separate spread parameters ($\overline{s}$, $\underline{s}$) and making no assumptions on uncertainty behavior, this version of the method is capable of obtaining precise intervals even when analyzing systems with complex uncertainty behavior (such as non-Gaussian or heteroscedastic noise). However, this increase in model robustness comes at the computational cost of having to train three times more parameters than a standard neural model.

This method has seen applications mainly in electricity load forecasting problems [2].

C. DIRECT METHODS

1) LOWER UPPER BOUND ESTIMATION (LUBE)

This method proposes a solution for PI construction where a single NN model with a modified architecture is directly trained to predict the upper and lower bounds of the interval [8].

The proposed network used for this method consists of a model similar to a traditional architecture, but where two units are used on the output layer. This means that some model parameters will be shared among both outputs, which makes them incompatible with a traditional backpropagation training routine. In order to solve this problem, multiple-output networks calculate the gradient of the shared weights as the mean between the gradient derived from each model output.

Afterwards, this method proposes to train the multiple-output architecture by using the Coverage Width-Based Criterion (CWC) equation presented in (61) as a loss function.

It is important to note that LUBE has consistently reported better and faster performance than all other sequential interval methods, which is why most of the recent investigation efforts dedicated to Neural PI models have been to improve the optimization procedure for the LUBE method. Since the CWC cost function is highly nonlinear, a wide range of non-conventional optimization algorithms have been tried, such as Simulated Annealing [8], Particle Swarm Optimization [66], and Genetic Algorithms [65], and even multi-objective evolutionary computation [67].

This method has seen applications in wind power forecasting [51], [66], [103], wind speed forecasting [104], [105], solar power forecasting [67], [106], electricity load forecasting [66], [107]–[109], streamflow discharge forecasting [110], landslide displacement prediction [111], reactor temperature prediction [112], [113], flood forecasting [114], [115] and electric arc furnace reactive power compensation estimation [116]

Reported variations for this method mainly consist of the inclusion of techniques such as additional feature selection [109], model ensembles [109], [112] and multi-objective optimization [117]. However, in [107] a different variation is proposed in the shape of a modified CWC cost function, where the PINAW term is replaced by a PI Normalized Root-mean-square Width (PINRW):

$$\text{PINRW} = \frac{1}{R}\sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\overline{y}(z_k) - \underline{y}(z_k)\right)^2} \tag{87}$$

where R is calculated as shown in equation (8). This modification was proposed to create an interval cost function which more closely resembles the behavior of a mean square error loss by imposing a heavier penalization on big forecasting errors.

Additionally, in [115] a different replacement for the PINAW term is proposed as Prediction Interval Average Relative Width (PIARW)

$$\text{PIARW} = \frac{1}{N}\sum_{k=1}^{N}\frac{\left(\overline{y}(z_k) - \underline{y}(z_k)\right)}{y_k} \tag{88}$$

where y_k is the target data for sample k. This work also proposes the inclusion of the Prediction Interval Symmetry (PIS) metric

$$\text{PIS} = \frac{1}{N}\sum_{k=1}^{N}\frac{\left|y_k - \frac{\overline{y}(z_k)+\underline{y}(z_k)}{2}\right|}{\overline{y}(z_k) - \underline{y}(z_k)} \tag{89}$$

so that the new Coverage Width Symmetry-based Criterion (CWSC) cost can be calculated as

$$\begin{aligned}\text{CWSC} = \gamma(\text{PIS})e^{\eta_3(\text{PIS}-\mu_2)} &+ \eta_2\text{PIARW} \\ &+\gamma(\text{PICP})e^{-\eta_1(\text{PICP}-\mu_1)}\end{aligned} \tag{90}$$

These modifications were proposed in an attempt to allow the LUBE method to make a more informed optimization by supplying full access to the training data points y_k on the cost function, while also giving importance to optimizing the model crisp prediction through the inclusion of the PIS component.

2) JOINT SUPERVISION

The Joint Supervision Method, as mentioned before in section III-C6 proposes the identification of one singular model composed by 3 outputs. This version of the method proposed in [64], uses a singular neural network as a base model, where two of its outputs will be used for the upper and lower interval bounds, while the third output will be associated to the predicted crisp value.

The training procedure follows the same steps described in section II-E for the fuzzy version of this method, solving the same cost functions presented in (45)-(47) for each NN output $(\overline{y}, \hat{y}, \underline{y})$.

The main advantage of the Joint Supervision Method over other Direct Methods is that the simplicity of its cost function allows the use of Stochastic Gradient Descent algorithms for optimization, which have been highly optimized for neural network training, resulting in much faster PI construction. Despite this decrease in computational cost, the inclusion of a hyperparameter λ, which must be optimized through a logarithmic search algorithm, into the model loss functions (see equations (45) and (47)) introduces the necessity of having to retrain the model several times in order to obtain an optimal solution. This is because, on each iteration of the λ hyperparameter logarithmic seach, a new Joint Supervision model must be trained using a different λ value for the cost function. This has a direct impact on model training times, which makes the Joint Supervision Method unable to fully take advantage of its simple cost function.

This method has seen applications mainly on load forecasting problems [64], [70].

3) QUALITY DRIVEN (QD) METHOD

The QD Method [7] proposed an improvement over the LUBE method through the introduction of a new, simplified cost function

$$L_{QD} = MPIW_{capt} + \lambda\frac{n}{\alpha(1-\alpha)}\max(0, (1-\alpha) - PICP)^2 \tag{91}$$

where λ is a hyperparameter that does not depend on the dataset nor the desired coverage probability, n is the total amount of data points, $(1-\alpha)$ is the desired PI coverage probability, and $MPIW_{capt}$ stands for Captured Mean Prediction Interval Width

$$MPIW_{capt} = \frac{\sum_{k=1}^{N}(\overline{y}(z_k) - \underline{y}(z_k))i_k}{\sum_{k=1}^{N} i_k} \tag{92}$$

where

$$i_k = \begin{cases} 1, & \text{if } \underline{y}(z_k) \leq y_k \leq \overline{y}(z_k) \\ 0, & \text{otherwise} \end{cases} \tag{93}$$

Using this procedure, the QD Method theoretically can build PIs using Stochastic Gradient Descent-based optimization, which makes it a faster alternative to LUBE while still retaining its interval quality. Additionally, the inclusion of the α parameter into the model cost function allows this method to explicitly introduce the desired coverage probability into the network training, which erases the need for a hyperparameter tuning procedure, unlike the Joint Supervision method. However, the performance of this method is limited because the proposed modifications to the LUBE cost function rely on the assumption that training data is independent and identically distributed, which is incompatible with dynamical system modeling.

Despite the reported advantages for this method, due to its novelty there are still no reported applications in literature which can corroborate experimental results with artificial datasets.

4) BAYES BY BACKPROP (BBB) METHOD

The BBB Method [118] proposes an algorithm that can be used to train Bayesian networks (where weights are modeled as probability distributions, as presented in the Bayesian Method) without having to resort to MCMC techniques and Hessian matrix calculations. This is achieved by proposing a modified backpropagation algorithm that is both compatible with Bayesian networks and can converge to a variational posterior of the weight distribution.

The BBB algorithm consists on applying the local reparameterization trick [119] to slightly alter the mean and standard deviation parameters of the weights on each iteration

$$\theta = (\mu, \sigma^2) \tag{94}$$

$$\epsilon \sim \mathcal{N}(0, 1) \tag{95}$$

$$f(\epsilon) = w = \mu + \sigma \cdot \epsilon \tag{96}$$

where θ is the vector of all network weights, μ is the vector of all network means, and σ is the vector of all network standard deviations. Using this technique, a gradient can be calculated and consequently the network parameters can be updated as

$$\nabla\mu = \frac{\partial f}{\partial w} + \frac{\partial f}{\partial \mu} \tag{97}$$

$$\nabla\sigma = \frac{\partial f}{\partial w}\frac{\epsilon}{\sigma} + \frac{\partial f}{\partial \sigma} \tag{98}$$

$$\mu \leftarrow (\mu - \alpha\nabla\mu) \tag{99}$$

$$\sigma \leftarrow (\sigma - \alpha\nabla\sigma) \tag{100}$$

Once the final posterior distribution is obtained, prediction uncertainty can be calculated by making several predictions for each input sample: Since network weights act as probability distributions, they are constantly reevaluated for each new prediction instance according to equation (96). Using this data, mean and standard deviation statistics can be calculated for the model prediction. Finally, a PI can be built by computing the corresponding quantile from the prediction distribution.

5) RANDOMIZED PRIOR FUNCTIONS METHOD

This method introduces a Bayesian approach for uncertainty estimation that does not require for network weights to be modeled as probability distributions [120]. Instead, the Randomized Prior Functions Method proposes a modified network architecture where total model predictions are obtained by adding the result of two separate networks: A regular, trainable network NN_{reg}, and a "prior", randomly-initialized non-trainable network NN_{prior}, so that

$$\hat{y}_{tot}(z_k) = \hat{y}_{reg}(z_k) + \hat{y}_{prior}(z_k) \tag{101}$$

where network NN_{prior} plays a role similar to the prior distribution defined in Bayesian inference, since it controls the model behavior in regions with higher uncertainty. This intuition is furtherly studied in the original article by proving that there is, in fact, a connection between generating a prediction using this architecture and sampling from the posterior distribution. This way, PIs can be estimated by first approximating the model posterior through repeated sampling by generating several model predictions, and then computing the corresponding quantile.

Although this method reported satisfactory results on its original publication [120], additional experiments are required to verify how its performance compares to other existing methods.

V. CHARACTERISTICS AND DISCUSSION OF METHODS

In this section, the main characteristics of the presented fuzzy and neural prediction intervals mentioned in sections III and IV are discussed.

In order to compare the performance and applicability of each of the presented methods, it is necessary to observe the main characteristics that distinguish each model, such as the methodology used to generate interval outputs, the number of parameters, the identification procedure, and the assumptions on the behavior of uncertainty. To facilitate this analysis, a summary of the main characteristics of each interval method is shown in Tables 2 for fuzzy sequential methods, 3 for fuzzy direct methods, 4 for neural sequential methods, and 5 for neural direct methods. In these tables, when specifying the number of parameters of each method, the notations presented in equations (5) and (49) were used.

For the case of interval methods based on fuzzy models, the number of parameters is affected by the structure of the base model. To perform a uniform quantification of this variable, an assumption is taken first. For the antecedents of the fuzzy models, fuzzy sets with Gaussian membership degrees are assumed to be used, thus for each input variables the mean and variance of the Gaussian distributions are considered as parameters. In summary, each local model has its own set of parameters for antecedents and consequences, thus the total quantity of parameters to be considered for both antecedents and consequences of the fuzzy model, increases proportionally with number of rules r.

It is important to note that, due to the different methodologies used by each model to quantify its computational cost, this variable was indirectly handled in the tables according to three criteria: The number of additional parameters introduced to the model structure by the interval method (computational cost increases with number of parameters), the type of optimization method that has to be used for parameter identification (gradient based methods have a lower computational cost than nonlinear alternatives), and the number of times a new model has to be trained to converge to an optimal solution (computational cost increases with the number of models and training repetitions).

TABLE 2. Characteristics of the fuzzy prediction intervals based on sequential methods.

Type of Interval	Characteristics
Covariance Method	- Method based on one previously identified TS model - Interval defined by 2 tuning parameters - Model training performed by a linear optimization method - No differences between internal and external sources of uncertainty (effects can be separated under some optional assumptions)
Fuzzy Number for Output Uncertainty	- Method based on one previously identified TS model - Interval defined by 2 additional parameters for antecedents and 2 additional parameters for consequences - Interval training performed by a fuzzy identification - Only external uncertainty is considered
Fuzzy Numbers (sequential version)	- Method based on one previously identified TS model - Interval defined by $2r(n_y + n_u + 1)$ additional parameters for consequences - Model training performed by a nonlinear optimization - No differences between internal and external sources of uncertainty
α-level cuts of the output	- Method based on one previously identified Mamdani model - Interval defined by one additional parameter (α) which set the bounds for consequences fuzzy sets - Model training performed by a nonlinear optimization - No differences between internal and external sources of uncertainty

Based on these tables, both fuzzy and neural direct methods tend to have a higher computational cost and number of parameters than sequential methods. This tendency can be explained since sequential methods rely mostly on a previous trained base model, so the training procedure is only applied for obtaining those parameters which are directly involved in the calculation of the PI. Instead, direct methods incorporate both the predictive model training and the interval model training into a single procedure, which can introduce some complexities (for example, in neural methods such as LUBE which cannot be trained with traditional neural network optimization algorithms, or fuzzy methods that consider larger quantities of parameters by including the concepts of type-2 fuzzy sets). The only exceptions among sequential methods to this tendency to have a lower computational cost than the direct methods, are those prediction intervals that do not rely on simplifications or assumptions on the behavior of uncertainty, such as the Bayesian Method and the Fuzzy Numbers Method.

Another aspect that can be noted is that the selection of the most suited interval model for any experiment depends on the characteristics of the application on which it will be used. Variables such as computational time available, uncertainty behavior (how many assumptions can be made without losing too much performance), and interval informativeness (how much useful information the interval provides) may be the more relevant when making this decision.

Additionally, among all reported interval methods, there exist three models which can be applied on both fuzzy and neural architectures: The covariance method, the fuzzy numbers method, and the joint supervision method. Although the main fundamentals are shared among both versions, some differences can still be appreciated on these methods. Regarding the covariance method, its main difference can be noted by observing equations (15), (16) and (82) where it can be seen that the fuzzy version of this method estimates prediction uncertainty on each of the fuzzy rules of the base model, and then calculates the total uncertainty as the weighted sum of each local estimation, while the neural version relies on a single calculation for uncertainty estimation. On the other hand, regarding the methods based on fuzzy numbers, its main difference lies in the type of parameters that are modeled as fuzzy numbers. On one hand, in the fuzzy version the method is applied to the consequence parameters, which means that the spread values can take advantage of the nonlinearities of the base model. On the other hand, the neural version only considers the use of fuzzy numbers for the weight values of the final layer. And the interval parameters cannot take advantage of the neural network nonlinearities since the output layer of neural networks considered by the methods does not use a nonlinear activation function. Finally, regarding the joint supervision method, its main difference can be appreciated in its training routine, where the neural method uses gradient descent-based methods which have been optimized for robust and efficient neural network training, while reported applications with the fuzzy method rely on evolutionary computation methods, that could increase its computational cost.

The following subsections will proceed to discuss the characteristics pertaining exclusively to methods based on fuzzy models, and later to methods based on neural models.

A. FUZZY PREDICTION INTERVALS

From the several fuzzy methods reported in this literature review, the versions based on type-2 fuzzy models and IT2 fuzzy neural networks stand out as the best suited methods for the identification and differentiation of the effects of internal and external sources of uncertainty. This is due to the structure of this kind of models, where part of the external uncertainty (mainly those associated to external inputs) can be represented by the use of type-2 fuzzy sets in the antecedents of the rules, while the internal uncertainty of the system can be handled by both, the rest of the type-2 antecedents and the type-1 consequences. Despite the fact that the other reported sequential methods cannot normally make this differentiation, some of them have the potential to separate their effect, such as the Covariance Method.

Considering the requirements of the specific application, the selection of the best suited interval method may vary. This decision depends on factors such as the type of data that is available and how detailed the estimations of the effects of internal and external uncertainty are desired. For example, if the model training data already contains interval measurements, the more suited alternative corresponds to the Interval-valued method, which is adapted for that type of data and then applies the ideas of interval arithmetic. On the other hand, if the application requires a detailed characterization of the effect of different types of uncertainty, the most appropriate choice corresponds to methods based on type-2 fuzzy models. Additionally, if the requirements of the user related

TABLE 3. Characteristics of the fuzzy prediction intervals based on direct methods.

Type of Interval	Characteristics
Min-Max Method	- Method based on two TS models - Antecedents defined by $2r(n_y + n_u)$ parameters - Consequences defined by $2r(n_y + n_u)$ parameters - Model training performed by a nonlinear optimization method - No differences between internal and external sources of uncertainty
Interval-Valued Data	- Method based on one TS model and interval arithmetic - Antecedents defined by $2r(2n_y + n_u)$ parameters - Consequences defined by $r(2n_y + n_u)$ parameters - Model training performed by a nonlinear optimization - Uncertainty of the output signal used as input of the model
Fuzzy Numbers (direct version)	- Method based on one TS model - Antecedents defined by $2r(n_y + n_u)$ parameters - Consequences defined by $2r(n_y + n_u + 1)$ parameters - Model training performed by nonlinear optimization - No differences between internal and external sources of uncertainty
Type-2 TS models	- Method based on one type-2 TS model - Antecedents defined by $3r(n_y + n_u)$ (assumption of $2r(n_y + n_u)$ membership functions with shared centers) - Consequences defined by $r(n_y + n_u)$ parameters in unnormalized A2-C0 model (+2 in normalized version) - Consequences defined by $2r(n_y + n_u)$ parameters in unnormalized A2-C1 model (+2 in normalized version) Model training performed by a nonlinear optimization - Effects of internal and external sources of uncertainty can be handled separately
IT2 Fuzzy Neural Networks	- Method based on one fuzzy neural network model - Antecedents defined by $3r(n_y + n_u)$ parameters (similar assumptions of type-2 TS models) - Consequences defined by $2r(n_y + n_u)$ parameters - Model training performed by nonlinear optimization - Effects of internal and external sources of uncertainty can be handled separately by the different layers of the network
Joint Supervision Method	- Method based on multiple identification of three TS models (varying one hyper-parameter) - Consequences defined by $3r(n_y + n_u)$ parameters - Model training performed by nonlinear optimization method - No differences between internal and external sources of uncertainty

TABLE 4. Characteristics of neural interval models based on sequential methods.

Type of Interval	Characteristics
Delta Method	- Method based on calculating network Jacobian and data noise variance. - Interval defined by 2 parameters $(\sigma_{tot}, t_{n-p}^{1-\frac{\alpha}{2}})$. - Model training performed by linear optimization methods. - Assumes uncertainty must be homogeneous and Gaussian.
Bayesian Method	- Method based on modeling network parameters as a probability distribution, represented by a mean and a variance. - Interval defined by $2(N_h(n_y + n_u + 1) + (N_h + 1))$ parameters. - Model training performed with MCMC methods and calculation of model Hessian matrix. - Makes no assumptions on uncertainty behavior.
Mean-Variance Estimation Method	- Method based on training two separate networks sequentially (one for point predictions, and one for error variance). - Interval defined by $(N_h(n_y + n_u + 1) + (N_h + 1))$ parameters. - Model training performed with Backpropagation methods. Variance network uses log-likelihood cost. - Assumes zero epistemic uncertainty.
Bootstrap Method	- Method based on training a bootstrap ensemble of neural networks. - Interval defined by $(B + 1)(N_h(n_y + n_u + 1) + (N_h + 1))$ parameters, where B is the bootstrap ensemble size. - Model training performed with backpropagation methods. - Assumes prediction error is symmetrical.
Covariance Method	- Method based on calculating regressor matrices, data noise variance, and a hyperparameter tuning procedure using a logarithmic search algorithm. - Interval defined by 2 parameters (α, σ_k). - Model training performed by linear optimization methods. - Assumes uncertainty is homogeneous and Gaussian.
Fuzzy Numbers Method	- Method based on modeling network parameters as fuzzy numbers with one mean and two spread values. - Interval defined by $2(N_h + 1)$ parameters. - Model training performed by nonlinear optimization methods. - Does not make assumptions on uncertainty behavior.

to detailed characterization are not as demanding, it is more recommended the use of the min-max method, the covariance method or the methods based on fuzzy numbers. This is because despite their simple structure, they are still useful for giving a general information of the total uncertainty present in the prediction of the modeled signal. In summary, below are included some general comments regarding the applicability of each method:

1) **Covariance Method:** This method is suitable for applications which require solutions with a low computational cost, where there are no external sources of uncertainty and system uncertainty can be reasonably approximated as homogeneous and Gaussian
2) **Fuzzy Numbers for Output Uncertainty:** The simplicity of the interval structure only allows its applicability to systems with additive external uncertainty.
3) **Fuzzy Numbers Method (sequential version):** The interval structure of this method is versatile and can be applicable to different type of systems and several uncertainty behaviors. Also, due to the use of a previous identified model, its training process does not increase in complexity.
4) **Method based on α-level cuts of the output:** This interval preserve the uncertainty information of the system without requiring additional computational effort for computing the uncertainty of the predicted output.
5) **Min-Max method:** There are more recent intervals which outperform the capacities of this method, such as the fuzzy numbers and the joint supervision methods. However, this method is still useful for cases where the uncertainty has a strong and persistent effect over time.
6) **Interval-Valued Data:** Its feasibility is restricted to applications where the dynamical system to be modeled presents interval-valued data as training inputs.
7) **Fuzzy Numbers Method (direct version):** Even though this method shares most of the advantages of its sequential counterpart, the training process of this version results more complex and computationally demanding since base model and interval identification are performed jointly at the same time.
8) **Type-2 TS models and IT2 Fuzzy Neural Networks:** Both methods are suitable for applications with complex system dynamics and uncertainty behaviors.

TABLE 5. Characteristics of neural interval models based on direct methods.

Type of Interval	Characteristics
LUBE Method	- Method based on training a single network with two outputs. - Interval defined by $N_h(n_y + n_u + 1) + 2(N_h + 1)$ parameters. - Model training performed by nonlinear optimization method. - Assumes prediction error is symmetrical.
Joint Supervision Method	- Method based on training a single network with three outputs, and a hyperparameter tuning procedure using a logarithmic search algorithm. - Interval defined by $N_h(n_y + n_u + 1) + 3(N_h + 1)$ parameters. - Model training performed with backpropagation methods. - Does not make assumptions on uncertainty behavior.
Quality-Driven Method	- Method based on training a single network with two outputs. - Interval defined by $N_h(n_y + n_u + 1) + 2(N_h + 1)$ parameters. - Model training performed with backpropagation methods. - Assumes prediction error is symmetrical
Bayes by Backprop Method	- Method based on modeling network parameters as a probability distribution, represented by a mean and variance - Interval defined by $2(N_h(n_y + n_u + 1) + (N_h + 1))$ parameters. - Model training performed with modified backpropagation methods. - Assumes prediction error is symmetrical
Randomized Prior Functions Method	- Method based on training a bootstrap ensemble of networks with an additional non-trainable prior network - Interval defined by $B(N_h(n_y + n_u + 1) + (N_h + 1))$ parameters, where B is the bootstrap ensemble size. - Model training performed with backpropagation methods. - Assumes prediction error is symmetrical

The structure of these models allows an exhaustive characterization of uncertainty in most of system dynamics. However, this comes at the disadvantage of an elevated computational cost for its training procedure.

9) **Joint Supervision Method:** Suitable for applications showcasing complex uncertainty behavior where the number of parameters and computational time are not a concern. This method can work well in most of dynamic systems, but in order to obtain good results, the three models which define the interval structure must be identified multiple times due to the search process of its hyperparameter.

Regarding its applications, fuzzy prediction interval models have been used on several types of problems. In this context, Table 6 shows a summary of the applications reported in literature for the methods described in this review, where cells left with a horizontal line ('—') represent cases where no literature examples were found for the given application-interval method combination.

As shown in Table 6, the uses of fuzzy prediction intervals are mainly focused on signal modeling in the context of renewable energy systems and operation of microgrids. There, the respective models were used for the formulation of robust control strategies. Other important applications which can be highlighted in the previous table is the modeling of nonlinear systems and chaotic time series forecasting. This is due to the extensive contribution of the IT2 fuzzy neural networks, which are widely researched in literature for this kind of applications.

B. NEURAL PREDICTION INTERVALS

Among the neural interval models presented in this review, the bootstrap method rises as an interesting proposal. This is because of two main reasons: First, the bootstrap method is the only method capable of estimating aleatory and epistemic uncertainty separately without any assumptions of homoscedasticity, which allows the obtention of unbiased estimators for these variables. Second, the bootstrap method can be used in conjunction with other interval models to furtherly improve their performance and/or quantify confidence bounds for the interval parameters.

Additionally, other notable interval methods that find utility in specific cases exist. For example, if an application requires very fast training times for online learning and the uncertainty of the model may be reasonably approximated as homoscedastic, the Delta Method will probably be the best suited solution. On the other hand, if an application does not call for online learning and presents a complex form of uncertainty that cannot be easily simplified, methods such as Fuzzy Numbers, Joint Supervision, Quality-Driven or even Bootstrap (especially if a separate quantification of aleatory and epistemic uncertainties is desired) may be the most appropriate solutions since they make none or very few assumptions on uncertainty behavior. In particular, the following comments can be made regarding the applicability of each method:

1) **Delta Method:** Suitable for applications that require very fast models with a small amount of parameters. Additionally, due to the assumptions of this method, system uncertainty must be studied to determine if it can be reasonably approximated as homogeneous and Gaussian.
2) **Bayesian Method:** The calculation of the Hessian matrix makes this method more suitable for applications where training times are not a problem. Additionally, the low amount of reported applications in literature for this method (in part due to the previously mentioned computational cost problems) makes its applicability harder to determine.
3) **Mean-Variance Estimation Method:** The assumption of zero epistemic uncertainty hinders the performance of this method, while other alternatives with similar computational costs, such as the Quality-Driven method, provide higher quality intervals since they do not rely on this assumption.
4) **Bootstrap Method:** Suitable for applications where long training times are not a limitation and a separation of epistemic and aleatory uncertainties is desired.
5) **Covariance Method:** While this method has a similar performance than the Delta Method, the necessity of performing a logarithmic search for hyperparameter optimization makes the Covariance Method more computationally demanding.

TABLE 6. Applications of the fuzzy prediction intervals.

Type of Interval Method	Applications			
	Fault detection algorithms	Renewable energy systems and micrograds	Non-linear systems and chaotic time series	Other Robust Control
Min-Max Method	[30]–[33]	[36]–[38]	[34]	[35]–[37]
Interval-Valued Data	—	—	[39]–[41]	—
Fuzzy Numbers (direct version)	—	[42]	—	—
Type-2 TS models	—	[23], [24], [50]–[53]	—	—
IT2 Fuzzy Neural Networks	—	[61]	[54]–[60] [62]	[60]
Covariance Method	[17]	[14], [20], [21]	[18], [19]	[16], [22]
Fuzzy Number for Output Uncertainty	—	[23], [24]	—	—
Fuzzy Numbers (sequential version)	—	[2]	—	—
α-level cuts of the output	—	[26]	[26]	—

6) **Fuzzy Numbers Method:** Since spread parameters are only used in the final network layer, the LUBE method produces higher quality intervals at the same computational cost.
7) **LUBE Method:** Suitable for applications showcasing complex uncertainty behavior where model weight and computational time are not a concern. Additionally, it is also recommended to perform a comparative analysis along with the Joint Supervision and Quality-Driven methods. In general, the LUBE method has a lower training time than the Joint Supervision method, while also not requiring any assumptions on uncertainty behavior.
8) **Joint Supervision Method:** Suitable for applications showcasing complex uncertainty behavior where model weight and computational time are not a concern. Additionally, it is also recommended to perform a comparative analysis along with the LUBE and Quality-Driven methods. In general, the Joint Supervision method has the highest training time among the 3 methods, but has the advantage of allowing non-symmetrical intervals, which can be useful for certain systems.
9) **Quality-Driven Method:** Suitable for applications showcasing complex uncertainty behavior where model weight and computational time are not a concern. Additionally, it is also recommended to perform a comparative analysis along with the LUBE and Joint Supervision methods. In general, the Quality-Driven Method has the shortest training time among the 3 methods, but relies on the assumption of i.i.d. data, which harm interval quality on dynamic systems.
10) **Bayes by Backprop Method:** Since no comparative experiments with other state of the art methods have been performed, this method is more suitable for research purposes.
11) **Randomized Prior Functions Method:** Since no comparative experiments with other state of the art methods have been performed, this method is more suitable for research purposes.

Additionally, reported model applications may also be used as comparison criteria for the models. In order to facilitate this analysis, a summary of the applications of each neural interval model covered in this work can be seen in Table 7. Similarly to table 6, cells left with a horizontal line ('—') represent cases where no literature examples were found for the given application-interval method combination.

As shown in Table 7, neural interval models have mainly seen use in robust control applications, especially on renewable energy systems and robust microgrid operation, where these models have been used for tasks such as wind speed forecasting, solar power prediction, and load forecasting.

VI. SIMULATION TESTS

In this section, comparisons between the most significant methods reported in previous sections are shown. Two benchmark cases were implemented for testing these methods: the modelling of the modified Chen series [121] and the forecasting of solar power generation data from Milan [122]. In order to make fair comparisons, all these methods were trained to comply with the 90% of coverage level requirement across the prediction horizon for the training database.

Among the fuzzy prediction interval models reported in section III, the methods selected for the simulation tests were: the sequential version of the interval based on fuzzy numbers, the fuzzy version of the joint supervision method and the unnormalized version of the Type-2 TSK fuzzy models. On the other hand, from the neural prediction interval

TABLE 7. Applications of neural prediction intervals.

Type of Interval Method	Applications			
	Fault detection algorithms	Renewable energy systems and microgrids	Non-linear systems and chaotic time series	Other Robust Control
Delta Method	[73]	[72]	[76]	[74], [75]
Bayesian Method	—	[82]	[76]	[83]
Mean-Variance Estimation Method	—	[85], [87]	[86], [88]	[88]
Bootstrap Method	[96]	[51], [92], [97]	[94]	[91], [93], [95], [98], [99]
Covariance Method	—	—	—	—
Fuzzy Numbers Method	—	[2]	—	—
Lower Upper Bound Estimation	[116]	[51], [66], [67], [103]–[109]	—	[110]–[115]
Joint Supervision Method	—	[64], [70]	—	—
Quality-Driven Method	—	—	—	—
Bayes by Backprop Method	—	—	—	—
Randomized Prior Functions Method	—	—	—	—

models reported in section IV, the methods selected were: the interval based on fuzzy numbers, the neural version of the joint supervision method and the quality driven method.

The simulation results for both benchmark cases considered for this work are shown below.

A. MODIFIED CHEN SERIES MODELING

The modified Chen series considered in this work has the same dynamics used in [64], which are described as:

$$\begin{aligned} y(k) = & \left[0.8 - 0.5\, exp\left\{-y^2(k-1)\right\}\right] y(k-1) \\ & - \left[0.3 - 0.9\, exp\left\{-y^2(k-1)\right\}\right] y(k-2) \\ & + u(k-1) + 0.2\, u(k-2) \\ & 0.1\, u(k-1)u(k-2) + e(k), \end{aligned} \tag{102}$$

where the noise depends on the previous state as follows

$$e(k) = 0.5\, exp\left\{-y(k-1)^2\right\} \beta(k), \tag{103}$$

and β is a white noise signal.

A total of 10.000 samples were obtained from the simulation of this series, which were separated into training, test and validation datasets, in proportions of 60%, 20% and 20% respectively. Based on this, fuzzy and neural network models, along with their respective intervals, were trained using the training set. On the other hand, the structure of both kind of models, which is defined by the quantity of rules (in the fuzzy approach), the number of layers and neurons (in the neural approach), and the selection of their input variables, was decided based on the minimization of the prediction error in the test dataset. Finally, the models and the corresponding prediction intervals were validated using the validation dataset, verifying that the trained intervals obtain a performance close to that obtained in the training set.

Due to the different ways of identifying the models and the interval of each type of method used, several architectures had to be considered. Tables 8 and 9 show a summary of the configurations that were used for each fuzzy and neural interval model, respectively. From these tables it can be noted that the obtained fuzzy models share most of their characteristics, with the difference being the fewer number of rules considered by the fuzzy joint supervision method when compared with the rest of fuzzy methods. This low number of rules, which is obtained during the structure optimization, results from the use of a different number of input variables and clustering algorithms, i.e., the fuzzy joint supervision method converges to a different structure (3 rules while using the Gustafson-Kessel clustering algorithm). On the

TABLE 8. Baseline architectures obtained for each fuzzy interval model on the modified Chen series dataset.

Hyperparameter	Fuzzy Models		
	Fuzzy Numbers	Joint Supervision	Type-2 TSK
Number of inputs	2 (autoregressors) 1 (exogenous)	2 (autoregressors) 2 (exogenous)	2 (autoregressors) 1 (exogenous)
Number of rules	18	3	18
Fuzzy clustering algorithm	Fuzzy C-Means	Gustafson Kessel	Fuzzy C-Means
Membership functions	Gaussian		
Type of model consequences	Linear		

TABLE 9. Baseline architectures obtained for each neural interval model on the modified Chen series dataset.

Hyperparameter	Neural Models		
	Fuzzy Numbers	Joint Supervision	Quality Driven
Number of inputs	2 (autoregressors) 2 (exogenous)	2 (autoregressors) 2 (exogenous)	2 (autoregressors) 2 (exogenous)
Number of hidden layer units	7	14	14
Optimizer	Levenberg-Marquardt	ADAM	ADAM
Learning rate	0.01	0.001	0.0001

other hand, the obtained neural models showcase more differences; while the Fuzzy Numbers network was trained using Levenberg-Marquardt optimization in order to replicate the model configuration reported in the source material [2], the Joint Supervision and Quality Driven models were trained using ADAM optimization. Additionally, the Quality Driven model was trained using a lower learning rate following the authors recommendations [7].

Table 10 shows the performance metrics obtained (RMSE, PICP and PINAW, defined in equations (6)-(10)) with the selected methods for one-, 8- and 16-prediction steps, all of them measured for the validation dataset. Additionally, the table includes the number of the parameters given by the crisp model (as defined above equation (2)) and the interval model. Also, Figures 3-8 present the prediction intervals obtained with the six selected methods for a 16-steps ahead prediction.

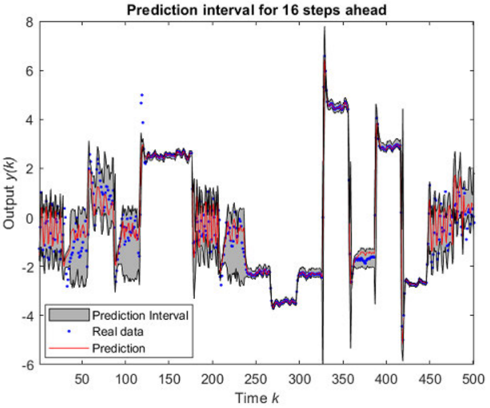


FIGURE 3. 16-steps ahead prediction interval based on fuzzy numbers.

As reported in Table 10, all methods achieve similar results of coverage level, close to the target value defined for training (PICP = 90%). This is expected due to the importance given to the PICP metric along all methods during the training procedure. An important difference observed among the methods is the interval width (PINAW) obtained along the prediction horizon. Despite the narrower intervals obtained by the methods based on neural networks when compared to those resulting from the methods that use fuzzy models for a one-step prediction horizon, the fuzzy numbers method (fuzzy version) and the type-2 fuzzy models show the best performances with the lowest values of PINAW for greater prediction steps. This better behavior along the prediction horizon can be explained because of the greater complexity of the interval structure associated to their large quantity of parameters, which allows a better fit of the interval to

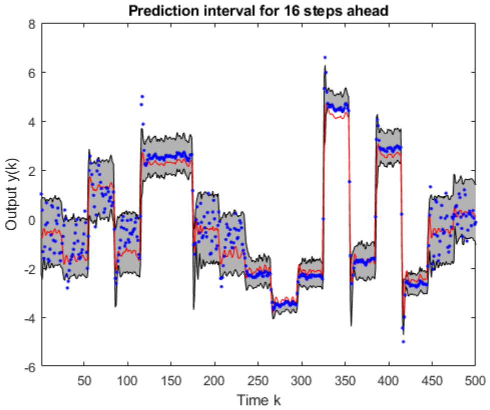


FIGURE 4. 16-steps ahead prediction interval based on fuzzy joint supervision.

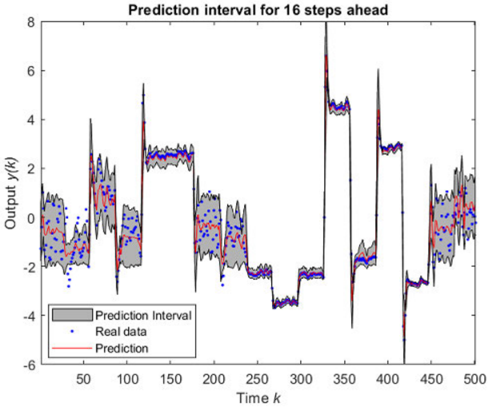


FIGURE 5. 16-steps ahead prediction interval based on type-2 fuzzy models.

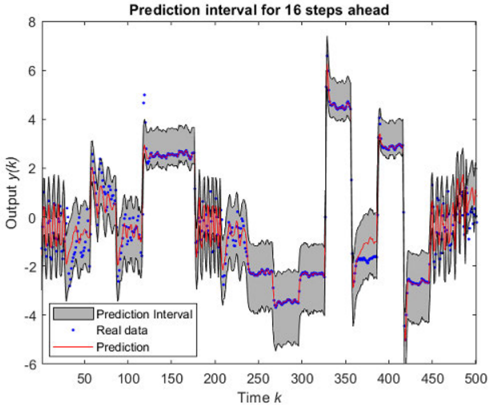


FIGURE 6. 16-steps ahead prediction interval based on neural fuzzy numbers.

TABLE 10. Performance metrics for the modified Chen series in the validation dataset.

Prediction Horizon	Metrics	Fuzzy Models			Neural Networks		
		Fuzzy Numbers	Joint Supervision	Type-2 TSK	Fuzzy Numbers	Joint Supervision	Quality Driven
1 step ahead	RMSE	0.243	0.332	0.252	0.238	0.241	0.252
	PICP (%)	89.14	88.79	88.14	88.69	90.03	91.09
	PINAW (%)	4.27	7.18	5.70	6.40	4.8	5.4
8 steps ahead	RMSE	0.534	0.61	0.495	0.511	0.593	0.605
	PICP (%)	89.35	88.7	90.36	90.11	92.81	90.5
	PINAW (%)	11.36	15.42	11.59	16.03	15.03	15.92
16 steps ahead	RMSE	0.618	0.65	0.521	0.617	0.627	0.662
	PICP (%)	89.86	91.33	91.17	90.62	89.3	90.21
	PINAW (%)	12.68	16.25	12.47	19.36	14.97	16.14
Number of Parameters	Crisp model	180	-	-	43	-	-
	Interval model	108	69	306	14	115	100

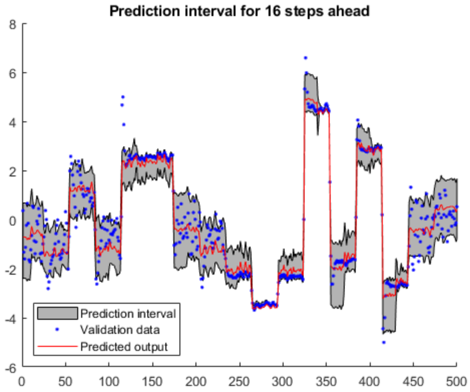


FIGURE 7. 16-steps ahead prediction interval based on neural joint supervision.

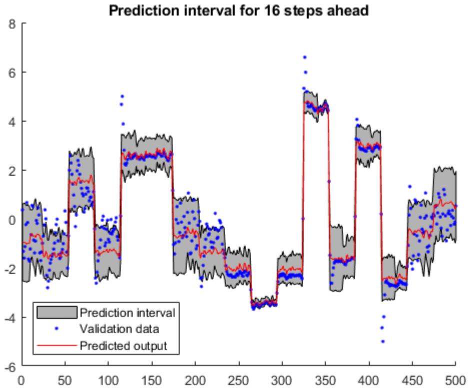


FIGURE 8. 16-steps ahead prediction interval based on quality driven method.

the accumulated uncertainty in the prediction. However, this greater complexity in the interval structure has the risk of causing convergence problems during the training process, due to the large degrees of freedom of the optimization problem that need to be solved.

It is important to highlight that for this case study, the type-2 fuzzy model presents the best results with respect to the other methods. This is achieved thanks to the fact that this method forces that a different model must be trained for each prediction step until the horizon is reached (for example, to calculate 8 steps ahead predictions, a total of 8 models must be trained, where each model incorporates the predictions of the previous steps models as inputs). This is an important difference of the type-2 fuzzy models with respect to other methods, which rely on only using a unique base model for all prediction steps. Additionally, as mentioned before, the large number of parameters provides a more complex structure that allows to reach better results than those obtained with the rest of direct methods implemented.

On the other hand, another interesting observation can be made by comparing the results obtained from the neural joint supervision and quality driven methods, where the joint supervision method achieves a slightly better performance. This can be attributed to the data independence assumptions that are made by the quality driven method: since dynamical systems are inherently dependent, the independence assumption can only be approximated via batch training. However, this technique is not perfect and could carry the risk of causing an overestimation of the error variance. On the other hand, the RMSE metric had lower values with the joint supervision method, probably due to the inclusion of an extra output unit trained specifically for the optimization of crisp predictions, while the training process of the quality driven method only focuses on optimizing the interval bounds.

Having discussed the results obtained for the Chen series modeling, the simulation results for the second benchmark case are presented next.

B. FORECASTING OF SOLAR POWER GENERATION

Solar power generation data from the Multi-Good Microgrid Laboratory (MG^2_{lab}) of the Politecnico di Milano, Milan, Italy [122], are used as the second case study for testing the prediction interval methods mentioned in this work. The available data corresponds to the measurements taken in the years 2017 and 2018, with a sample time of one hour. A total of 11.688 samples are divided in the same proportions as the previous case study: 60% for training, 20% for testing and 20% for validation.

The training process of the intervals are performed using a target value for the coverage level of 90%. Similar to the

Chen series experiment, the structure optimization of the models was decided based on the configuration that had the best performance of the metrics in the test dataset. A total of 48 regressors, corresponding to the data of two previous days, were used as the initial candidate inputs, selecting the most relevant inputs by performing a sensitivity analysis.

Tables 11 and 12 show a summary of the different baseline configurations that were obtained for each fuzzy and neural interval model, respectively.

TABLE 11. Baseline architectures obtained for each fuzzy interval model.

Hyperparameter	Fuzzy Models		
	Fuzzy Numbers	Joint Supervision	Type-2 TSK
Number of inputs	23	23	23
Number of rules	2	2	2
Fuzzy clustering algorithm	Fuzzy C-Means	Fuzzy C-Means	Fuzzy C-Means
Membership functions	Gaussian		
Type of model consequences	Linear		

TABLE 12. Baseline architectures obtained for each neural interval model.

Hyperparameter	Neural Models		
	Fuzzy Numbers	Joint Supervision	Quality Driven
Number of inputs	17	31	31
Number of hidden layer units	5	33	33
Optimizer	Levenberg-Marquardt	ADAM	ADAM
Learning rate	0.01	0.001	0.0001

From tables 11 and 12, it can be noted that, unlike the previous case of the modified Chen series experiment, in this case all fuzzy models reported on Table 11 share the same hyperparameters. This is because this time the same number of input variables and clustering algorithms were used. On the other hand, the neural models reported on Table 12 preserved most of their hyperparameters that were used on the case of the modified Chen series experiment.

Table 13, shows the performance metrics obtained with the selected methods for different prediction horizons, all of them measured on the validation dataset. Additionally, Figures 9-14 present the prediction intervals obtained with the selected methods for a 24 hours ahead prediction.

From Table 13, similar tendencies can be appreciated on some models when compared to the results obtained for the Chen series modeling. In these results, all methods managed to comply with the desired coverage level (PICP) of 90%, while the fuzzy prediction intervals based on fuzzy numbers, the type-2 models and the quality driven method presented the overall narrowest intervals throughout the prediction horizon (see PINAW in Table 13). Additionally, from Figures 9-12 it can be seen that the fuzzy prediction interval based on fuzzy numbers presents a smoother behavior throughout the plotted data.

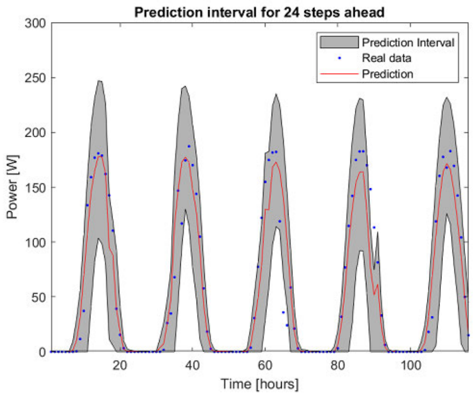


FIGURE 9. 24-steps ahead prediction interval based on fuzzy numbers.

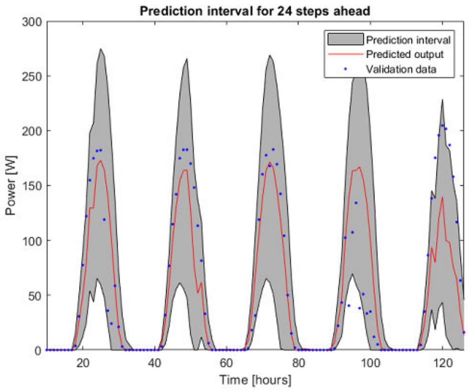


FIGURE 10. 24-steps ahead prediction interval based on fuzzy joint supervision.

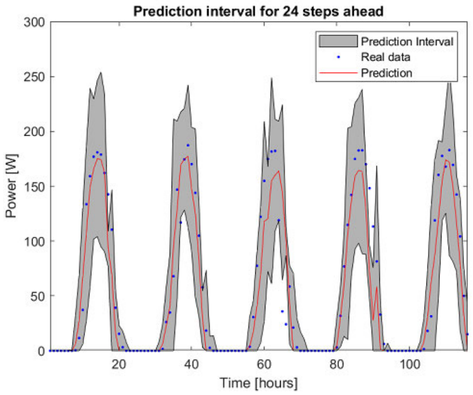


FIGURE 11. 24-steps ahead prediction interval based on type-2 fuzzy models.

Since the explanation of the main tendencies presented by the intervals were already covered in the previous benchmark case, here the discussion will be focused on those results that did not follow these previous trends.

TABLE 13. Performance metrics in the validation dataset.

Prediction Horizon	Metrics	Fuzzy Models			Neural Networks		
		Fuzzy Numbers	Joint Supervision	Type-2 TSK	Fuzzy Numbers	Joint Supervision	Quality Driven
1 hour ahead	RMSE	13.48	14.51	13.85	12.70	12.70	13.02
	PICP (%)	89.65	90.57	89.05	91.24	93.53	91.53
	PINAW (%)	8.62	18.47	10.55	9.93	12.97	9.64
12 hours ahead	RMSE	25.95	26.01	27.01	25.31	25.29	27.69
	PICP (%)	89.08	90.17	89.04	91.62	91.79	91.00
	PINAW (%)	17.83	28.45	20.36	20.61	23.23	22.69
24 hours ahead	RMSE	25.91	26.02	27.37	25.07	25.45	25.13
	PICP (%)	88.76	91.13	88.63	91.84	91.09	91.17
	PINAW (%)	17.44	26.03	20.31	20.78	21.25	19.30
Number of Parameters	Crisp model	140	-	-	96	-	-
	Interval model	92	236	234	10	1158	1124

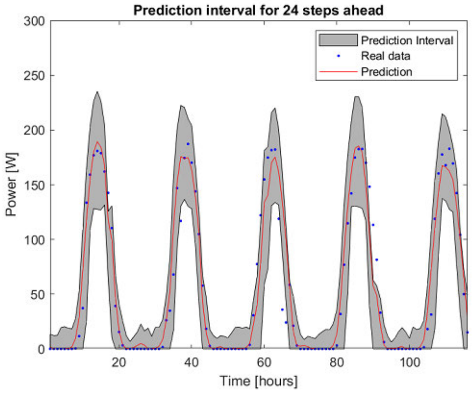


FIGURE 12. 24-steps ahead prediction interval based on neural fuzzy numbers.

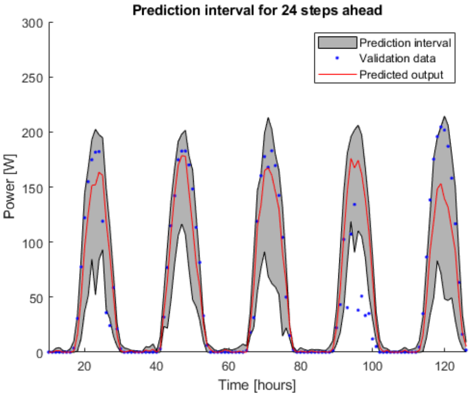


FIGURE 13. 24-steps ahead prediction interval based on neural joint supervision.

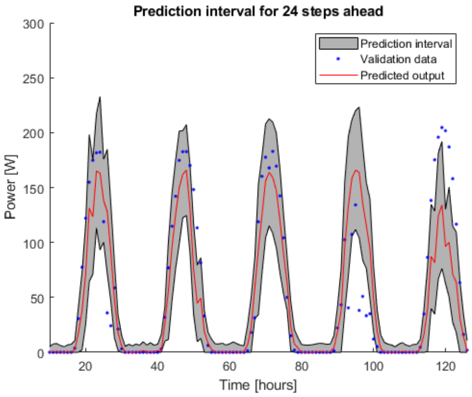


FIGURE 14. 24-steps ahead prediction interval based on quality driven method.

When compared with the Chen series results, the direct methods based on neural networks obtained wider intervals, with a coverage level greater than the target value defined for training, especially the Joint Supervision method. The main challenge when training predictive models for solar power forecasting lies in dealing with a hard lower bound at zero power which also appears in 50-60% of the dataset (corresponding to night hours). This phenomenon can impact the model performance in two main ways: First, if model predictions are not handled properly, it is possible to obtain negative values for the predictions. Since no negative data is present on the dataset, these predictions can produce an abnormal increment of the uncertainty estimation. This situation heavily impacts interval quality, specifically for larger prediction steps, which are affected by the uncertainty accumulation of its previous predictions. Secondly, since an important proportion of the dataset is zero-valued, the resulting search space for the optimizer is highly unstable, which can produce bad local minima or exploding gradients on neural models. However, if correct preprocessing and postprocessing measures are taken, it can be appreciated that average interval width tends to abruptly descend on these regions. This is because samples corresponding to night hours intrinsically have a lower uncertainty due to the absence of solar radiation.

Despite all the previous considerations, it must still be noted that in Table 13 the Joint Supervision method presents a higher drop in performance in comparison to the Quality Driven model even though both methods should be equally affected by the complications of the dataset. This phenomenon can instead be attributed to the logarithmic search used by the Joint Supervision method for hyperparameter optimization, since this process is highly vulnerable to parameter initialization uncertainty and can sometimes present convergence problems on complex solution spaces, such as in this experiment.

To finish the analysis and comments of the results obtained by both benchmark cases studied in this work, it is important to remark the following situation presented during the training process of the intervals. Due to the nonlinear characteristic of the optimization problem solved during training, along with their large number of high degrees of freedom (large number of optimization variables), the implementation of all methods suffer of a high probability to fall into a suboptimal solution. This problem can be mitigated by guiding the optimal search and reducing the space of possible solutions. Another option that can be applied is a fine-tuning of the hyperparameters which define the training algorithm. As an example of this, it can be mentioned that the first strategy of mitigation was applied in the training process of the fuzzy numbers method based on fuzzy models. The results obtained for both simulated cases shown that this method based on fuzzy numbers stands out for its performance over the rest of methods, achieving narrow interval widths. This good behavior was achieved after testing bad previous local minima obtained with optimizations that had greater spaces of possible solutions. This shows the importance of considering the use of these mitigation techniques, to achieve convergence to an optimal prediction interval model during the training procedure.

VII. CONCLUSION

A detailed literature review of the methods used for the development of prediction intervals based on fuzzy logic systems and neural networks has been made in this paper. Moreover, the main characteristics of each method for prediction interval construction were presented and some recommendations were made for selecting the most suited method for specific applications. As a final contribution, this work also performed experimental tests to compare the performance of the most significant methods reported in this review, using benchmark cases that consisted of artificial and real data.

The importance of characterizing the uncertainty associated to signals and predictions is evidenced in this work, given its applicability to different engineering problems. It has been highlighted that the methods presented in this work, corresponding to the prediction intervals based on fuzzy models and neural networks, are widely used in the literature for the implementation of forecasting systems, fault detection algorithms, robust control strategies and the study of the future behavior of nonlinear systems.

REFERENCES

[1] Y. Xu, M. Zhang, Q. Zhu, and Y. He, "An improved multi-Kernel RVM integrated with CEEMD for high-quality intervals prediction construction and its intelligent modeling application," *Chemometric Intell. Lab. Syst.*, vol. 171, pp. 151–160, Dec. 2017.

[2] L. G. Marín, N. Cruz, D. Sáez, M. Sumner, and A. Núñez, "Prediction interval methodology based on fuzzy numbers and its extension to fuzzy systems and neural networks," *Expert Syst. Appl.*, vol. 119, pp. 128–141, Apr. 2019.

[3] H. M. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Neural network-based uncertainty quantification: A survey of methodologies and applications," *IEEE Access*, vol. 6, pp. 36218–36234, 2018.

[4] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," *IEEE Trans. Neural Netw.*, vol. 22, no. 9, pp. 1341–1356, Sep. 2011.

[5] T. Heskes, "Practical confidence and prediction intervals," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 176–182.

[6] G. Papadopoulos, P. J. Edwards, and A. F. Murray, "Confidence estimation methods for neural networks: A practical comparison," *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1278–1287, Nov. 2001.

[7] T. Pearce, A. Brintrup, M. Zaki, and A. Neely, "High-quality prediction intervals for deep learning: A distribution-free, ensembled approach," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2018, pp. 4075–4084.

[8] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 337–346, Mar. 2011.

[9] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.

[10] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 535–550, Oct. 2000.

[11] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Inf. Sci.*, vol. 8, no. 3, pp. 199–249, May 1975.

[12] J. J. Buckley and Y. Hayashi, "Fuzzy neural networks: A survey," *Fuzzy Sets Syst.*, vol. 66, no. 1, pp. 1–13, Aug. 1994.

[13] P. V. de Campos Souza, "Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature," *Appl. Soft Comput.*, vol. 92, Jul. 2020, Art. no. 106275.

[14] D. Saez, F. Avila, D. Olivares, C. Canizares, and L. Marin, "Fuzzy prediction interval models for forecasting renewable resources and loads in microgrids," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 548–556, Mar. 2015.

[15] I. Škrjanc, "Fuzzy confidence interval for pH titration curve," *Appl. Math. Model.*, vol. 35, no. 8, pp. 4083–4090, Aug. 2011.

[16] A. Bayas, I. Škrjanc, and D. Sáez, "Design of fuzzy robust control strategies for a distributed solar collector field," *Appl. Soft Comput.*, vol. 71, pp. 1009–1019, Oct. 2018.

[17] M. Crispoltoni, M. Fravolini, F. Balzano, S. D'Urso, and M. Napolitano, "Interval fuzzy model for robust aircraft IMU sensors fault detection," *Sensors*, vol. 18, no. 8, p. 2488, Aug. 2018.

[18] S. Tomazic, D. Dovzan, and I. Skrjanc, "Confidence-interval-fuzzy-model-based indoor localization," *IEEE Trans. Ind. Electron.*, vol. 66, no. 3, pp. 2015–2024, Mar. 2019.

[19] A. Nunez and B. De Schutter, "Distributed identification of fuzzy confidence intervals for traffic measurements," in *Proc. IEEE 51st IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 6995–7000.

[20] R. Morales, D. Saez, L. G. Marin, and A. Nunez, "Microgrid planning based on fuzzy interval models of renewable resources," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2016, pp. 336–343.

[21] S. F. Rafique, Z. Jianhua, R. Rafique, J. Guo, and I. Jamil, "Renewable generation (wind/solar) and load modeling through modified fuzzy prediction interval," *Int. J. Photoenergy*, vol. 2018, pp. 1–14, Jan. 2018.

[22] O. Cartagena, D. Munoz-Carpintero, and D. Saez, "A robust predictive control strategy for building HVAC systems based on interval fuzzy models," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2018, pp. 1–8.

[23] M. S. Fadali, S. Jafarzadeh, and A. Nafeh, "Fuzzy TSK approximation using type-2 fuzzy logic systems and its application to modeling a photovoltaic array," in *Proc. Amer. Control Conf.*, Jun. 2010, pp. 6454–6459.

[24] S. Jafarzadeh, M. S. Fadali, and C. Y. Evrenosoglu, "Solar power prediction using interval Type-2 TSK modeling," *IEEE Trans. Sustain. Energy*, vol. 4, no. 2, pp. 333–339, Apr. 2013.

[25] N. N. Karnik and J. M. Mendel, "Operations on type-2 fuzzy sets," *Fuzzy Sets Syst.*, vol. 122, no. 2, pp. 327–348, Sep. 2001.

[26] D. Pekaslan, C. Wagner, J. M. Garibaldi, L. G. Marin, and D. Saez, "Uncertainty-aware forecasting of renewable energy sources," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2020, pp. 240–246.

[27] L.-X. Wang, *A Course in Fuzzy Systems and Control*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[28] H. Hamrawi, S. Coupland, and R. John, "Type-2 fuzzy alpha-cuts," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 3, pp. 682–692, Jun. 2017.

[29] I. Škrjanc, S. Blažič, and O. Agamennoni, "Identification of dynamical systems with a robust interval fuzzy model," *Automatica*, vol. 41, no. 2, pp. 327–332, Feb. 2005.

[30] S. Oblak, "Interval fuzzy modelling in fault detection for a class of processes with interval-type parameters," in *Proc. EUROCON Int. Conf. Comput. Tool*, Nov. 2005, pp. 1867–1870.

[31] S. Oblak, I. Skrjanc, and S. Blazic, "On applying interval fuzzy model to fault detection and isolation for nonlinear input-output systems with uncertain parameters," in *Proc. IEEE Conf. Control Appl. CCA*, Aug. 2005, pp. 465–470.

[32] S. Oblak, I. Škrjanc, and S. Blažič, "Fault detection for nonlinear systems with uncertain parameters based on the interval fuzzy model," *Eng. Appl. Artif. Intell.*, vol. 20, no. 4, pp. 503–510, Jun. 2007.

[33] S. Oblak, I. Skrjanc, and S. Blazic, "A new fault-detection system for nonlinear systems based on an interval fuzzy model," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2007, pp. 2956–2962.

[34] Y.-W. Chen, J.-B. Yang, C.-C. Pan, D.-L. Xu, and Z.-J. Zhou, "Identification of uncertain nonlinear systems: Constructing belief rule-based models," *Knowl.-Based Syst.*, vol. 73, pp. 124–133, Jan. 2015.

[35] D. Senthilkumar and C. Mahanta, "Identification of uncertain nonlinear systems for robust fuzzy control," *ISA Trans.*, vol. 49, no. 1, pp. 27–38, Jan. 2010.

[36] F. Valencia, J. Collado, D. Saez, and L. G. Marin, "Robust energy management system for a microgrid based on a fuzzy prediction interval model," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1486–1494, May 2016.

[37] F. Valencia, D. Saez, J. Collado, F. Avila, A. Marquez, and J. J. Espinosa, "Robust energy management system based on interval fuzzy models," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 1, pp. 140–157, Jan. 2016.

[38] N. Thao and K. Uchida, "An improved interval fuzzy modeling method: Applications to the estimation of photovoltaic/wind/battery power in renewable energy systems," *Energies*, vol. 11, no. 3, p. 482, Feb. 2018.

[39] Z. Xu and C. Sun, "Interval T-S fuzzy model and its application to identification of nonlinear interval dynamic system based on interval data," in *Proc. 48th IEEE Conf. Decis. Control (CDC) Held Jointly 28th Chin. Control Conf.*, Dec. 2009, pp. 4144–4149.

[40] S. Zaidi and A. Kroll, "A novel approach to T-S fuzzy modeling of nonlinear dynamic systems with uncertainties using symbolic interval-valued outputs," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 1196–1201, 2015.

[41] S. Zaidi and A. Kroll, "NOE TS fuzzy modelling of nonlinear dynamic systems with uncertainties using symbolic interval-valued data," *Appl. Soft Comput.*, vol. 57, pp. 353–362, Aug. 2017.

[42] F. Veltman, L. G. Marin, D. Saez, L. Guitierrez, and A. Nunez, "Prediction interval modeling tuned by an improved teaching learning algorithm applied to load forecasting in microgrids," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Dec. 2015, pp. 651–658.

[43] J. M. Mendel, *Interval Type-2 Fuzzy Systems*. Cham, Switzerland: Springer, 2017, pp. 449–527.

[44] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Inf. Sci.*, vol. 132, nos. 1–4, pp. 195–220, Feb. 2001.

[45] N. N. Karnik and J. M. Mendel, "Applications of type-2 fuzzy logic systems to forecasting of time-series," *Inf. Sci.*, vol. 120, nos. 1–4, pp. 89–111, Nov. 1999.

[46] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems," in *Proc. 9th IEEE Int. Conf. Fuzzy Syst. FUZZ*, Oct. 2000, pp. 328–333.

[47] K. Huarng and H.-K. Yu, "A type 2 fuzzy time series model for stock index forecasting," *Phys. A, Stat. Mech. Appl.*, vol. 353, pp. 445–462, Aug. 2005.

[48] A. Khosravi, S. Nahavandi, and D. Creighton, "Short term load forecasting using interval Type-2 fuzzy logic systems," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jun. 2011, pp. 502–508.

[49] A. Khosravi, S. Nahavandi, D. Creighton, and D. Srinivasan, "Interval type-2 fuzzy logic systems for load forecasting: A comparative study," *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1274–1282, Aug. 2012.

[50] A. Khosravi and S. Nahavandi, "Combined nonparametric prediction intervals for wind power generation," *IEEE Trans. Sustain. Energy*, vol. 4, no. 4, pp. 849–856, Oct. 2013.

[51] A. Khosravi, S. Nahavandi, and D. Creighton, "Prediction intervals for short-term wind farm power generation forecasts," *IEEE Trans. Sustain. Energy*, vol. 4, no. 3, pp. 602–610, Jul. 2013.

[52] L. G. Marin, F. Valencia, and D. Saez, "Prediction interval based on type-2 fuzzy systems for wind power generation and loads in microgrid control design," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2016, pp. 328–335.

[53] W. Zou, C. Li, and P. Chen, "An inter Type-2 FCR algorithm based T–S fuzzy model for short-term wind power interval prediction," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 4934–4943, Sep. 2019.

[54] C.-F. Juang and Y.-W. Tsao, "A self-evolving interval Type-2 fuzzy neural network with online structure and parameter learning," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 6, pp. 1411–1424, Dec. 2008.

[55] C.-F. Juang and R.-B. Huang, "A mamdani-type recurrent interval Type-2 fuzzy neural network for identification of dynamic systems with measurement noise," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 8975–8980, Jan. 2011.

[56] Y.-Y. Lin, S.-H. Liao, J.-Y. Chang, and C.-T. Lin, "Simplified interval Type-2 fuzzy neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 959–969, May 2014.

[57] V. Sumati, P. Chellapilla, S. Paul, and L. Singh, "Parallel interval Type-2 subsethood neural fuzzy inference system," *Expert Syst. Appl.*, vol. 60, pp. 156–168, Oct. 2016.

[58] N. Baklouti, A. Abraham, and A. M. Alimi, "A beta basis function interval type-2 fuzzy neural network for time series applications," *Eng. Appl. Artif. Intell.*, vol. 71, pp. 259–274, May 2018.

[59] N. Anh, S. Suresh, M. Pratama, and N. Srikanth, "Interval prediction of wave energy characteristics using meta-cognitive interval type-2 fuzzy inference system," *Knowl.-Based Syst.*, vol. 169, pp. 28–38, Apr. 2019.

[60] C.-M. Lin and T.-L. Le, "PSO-self-organizing interval Type-2 fuzzy neural network for antilock braking systems," *Int. J. Fuzzy Syst.*, vol. 19, no. 5, pp. 1362–1374, Oct. 2017.

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

[62] M. Han, K. Zhong, T. Qiu, and B. Han, "Interval type-2 fuzzy neural networks for chaotic time series prediction: A concise overview," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2720–2731, Jul. 2019.

[63] D. Munoz-Carpintero, S. Parra, O. Cartagena, D. Saez, L. G. Marin, and I. Skrjanc, "Fuzzy interval modelling based on joint supervision," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2020, pp. 1–8.

[64] N. Cruz, L. G. Marin, and D. Saez, "Neural network prediction interval based on joint supervision," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[65] R. Ak, Y. LI, and E. Zio, "Estimation of prediction intervals of neural network models by a multi-objective genetic algorithm," in *Uncertainty Modeling in Knowledge Engineering and Decision Making*. Singapore: World Scientific, 2012, pp. 1036–1041.

[66] H. Quan, D. Srinivasan, and A. Khosravi, "Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 303–315, Feb. 2014.

[67] I. M. Galván, J. M. Valls, A. Cervantes, and R. Aler, "Multi-objective evolutionary optimization of prediction intervals for solar energy forecasting with neural networks," *Inf. Sci.*, vols. 418–419, pp. 363–382, Dec. 2017.

[68] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[69] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[70] N. Cruz, L. G. Marin, and D. Saez, "Prediction intervals with LSTM networks trained by joint supervision," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[71] L. H. Ungar, R. D. De Veaux, and E. Rosengarten, "Estimating prediction intervals for artificial neural networks," in *Proc. 9th Yale Workshop Adapt. Learn. Syst.*, 1996, pp. 1–6.

[72] J. T. G. Hwang and A. A. Ding, "Prediction intervals for artificial neural networks," *J. Amer. Statist. Assoc.*, vol. 92, no. 438, pp. 748–757, Jun. 1997.

[73] S. L. Ho, M. Xie, L. C. Tang, K. Xu, and T. N. Goh, "Neural network modeling with confidence bounds: A case study on the solder paste deposition process," *IEEE Trans. Electron. Packag. Manuf.*, vol. 24, no. 4, pp. 323–332, Oct. 2001.

[74] T. Lu and M. Viljanen, "Prediction of indoor temperature and relative humidity using neural network models: Model comparison," *Neural Comput. Appl.*, vol. 18, no. 4, p. 345, 2009.

[75] Y.-M. Kuo, C.-W. Liu, and K.-H. Lin, "Evaluation of the ability of an artificial neural network model to assess the variation of groundwater quality in an area of blackfoot disease in taiwan," *Water Res.*, vol. 38, no. 1, pp. 148–158, Jan. 2004.

[76] A. Khosravi, E. Mazloumi, S. Nahavandi, D. Creighton, and J. W. C. van Lint, "Prediction intervals to account for uncertainties in travel time prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 537–547, Jun. 2011.

[77] R. D. De VlEAUX, J. Schumi, J. Schweinsberg, and L. H. Ungar, "Prediction intervals for neural networks via nonlinear regression," *Technometrics*, vol. 40, no. 4, pp. 273–282, Nov. 1998.

[78] A. Khosravi, S. Nahavandi, and D. Creighton, "A prediction interval-based approach to determine optimal structures of neural network metamodels," *Expert Syst. Appl.*, vol. 37, no. 3, pp. 2377–2387, Mar. 2010.

[79] A. Khosravi, S. Nahavandi, and D. Creighton, "Construction of optimal prediction intervals for load forecasting problems," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1496–1503, Aug. 2010.

[80] J. S. Denker and Y. LeCun, "Transforming neural-net output levels to probability distributions," in *Proc. Adv. Neural Inf. Process. Syst.*, 1991, pp. 853–859.

[81] D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, May 1992.

[82] A. Khosravi, S. Nahavandi, and D. Creighton, "Load forecasting and neural networks: A prediction interval-based perspective," in *Computational Intelligence in Power Engineering* (Studies in Computational Intelligence), vol. 302. Berlin, Germany: Springer-Verlag, 2010, pp. 131–150, doi: 10.1007/978-3-642-14013-6_5.

[83] A. F. Torres, W. R. Walker, and M. McKee, "Forecasting daily potential evapotranspiration using machine learning and limited climatic data," *Agricult. Water Manage.*, vol. 98, no. 4, pp. 553–562, Feb. 2011.

[84] D. A. Nix and A. S. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proc. IEEE Int. Conf. Neural Netw. (ICNN)*, Jun. 1994, pp. 55–60.

[85] A. Khosravi and S. Nahavandi, "An optimized mean variance estimation method for uncertainty quantification of wind power forecasts," *Int. J. Electr. Power Energy Syst.*, vol. 61, pp. 446–454, Oct. 2014.

[86] J.-N. Hwang and E. Little, "Real time recurrent neural networks for time series prediction and confidence estimation," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Jun. 1996, pp. 1889–1894.

[87] M. Rigamonti, P. Baraldi, E. Zio, I. Roychoudhury, K. Goebel, and S. Poll, "Ensemble of optimized echo state networks for remaining useful life prediction," *Neurocomputing*, vol. 281, pp. 121–138, Mar. 2018.

[88] W. Yao, Z. Zeng, and C. Lian, "Generating probabilistic predictions using mean-variance estimation and echo state network," *Neurocomputing*, vol. 219, pp. 536–547, Jan. 2017.

[89] G. Paass, "Assessing and improving neural network predictions by the bootstrap algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 1993, pp. 196–203.

[90] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artif. Intell.*, vol. 137, nos. 1–2, pp. 239–263, May 2002.

[91] M. R. A. van Gent, H. F. P. van den Boogaard, B. Pozueta, and J. R. Medina, "Neural network modelling of wave overtopping at coastal structures," *Coastal Eng.*, vol. 54, no. 8, pp. 586–593, Aug. 2007.

[92] P. Pinson and G. Kariniotakis, "On-line assessment of prediction risk for wind power production forecasts," *Wind Energy*, vol. 7, no. 2, pp. 119–132, Apr. 2004.

[93] M. Talebizadeh and A. Moridnejad, "Uncertainty analysis for the forecast of lake level fluctuations using ensembles of ANN and ANFIS models," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 4126–4135, Apr. 2011.

[94] E. Mazloumi, G. Rose, G. Currie, and S. Moridpour, "Prediction intervals to account for uncertainties in neural network predictions: Methodology and application in bus travel time prediction," *Eng. Appl. Artif. Intell.*, vol. 24, no. 3, pp. 534–542, Apr. 2011.

[95] E. Zio, "A study of the bootstrap method for estimating the accuracy of artificial neural networks in predicting nuclear transient processes," *IEEE Trans. Nucl. Sci.*, vol. 53, no. 3, pp. 1460–1478, Jun. 2006.

[96] P. Baraldi, M. Compare, S. Sauco, and E. Zio, "Ensemble neural network-based particle filtering for prognostics," *Mech. Syst. Signal Process.*, vol. 41, nos. 1–2, pp. 288–300, Dec. 2013.

[97] A. Khosravi, S. Nahavandi, and D. Creighton, "Quantifying uncertainties of neural network-based electricity price forecasts," *Appl. Energy*, vol. 112, pp. 120–129, Dec. 2013.

[98] B. D. Conduit, N. G. Jones, H. J. Stone, and G. J. Conduit, "Design of a nickel-base superalloy using a neural network," *Mater. Design*, vol. 131, pp. 358–365, Oct. 2017.

[99] K. Ristovski, S. Vucetic, and Z. Obradovic, "Uncertainty analysis of Neural-Network-Based aerosol retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 2, pp. 409–414, Feb. 2012.

[100] J. G. Carney, P. Cunningham, and U. Bhagwan, "Confidence and prediction intervals for neural network ensembles," in *Proc. IJCNN Int. Joint Conf. Neural Netw.*, Jul. 1999, pp. 1215–1218.

[101] A. Khosravi, S. Nahavandi, D. Creighton, and D. Srinivasan, "Optimizing the quality of bootstrap-based prediction intervals," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 3072–3078.

[102] L. G. Marín, M. Sumner, D. Mu noz-Carpintero, D. Köbrich, S. Pholboon, D. Sáez, and A. Núñez, "Hierarchical energy management system based on fuzzy prediction intervals for operation and coordination of microgrids," *Energies*, vol. 12, no. 4453, 2019.

[103] Z. Shi, H. Liang, and V. Dinavahi, "Direct interval forecast of uncertain wind power based on recurrent neural networks," *IEEE Trans. Sustain. Energy*, vol. 9, no. 3, pp. 1177–1187, Jul. 2018.

[104] R. Li and Y. Jin, "A wind speed interval prediction system based on multi-objective optimization for machine learning method," *Appl. Energy*, vol. 228, pp. 2207–2220, Oct. 2018.

[105] 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," Tech. Rep., Sep. 2013.

[106] Q. Ni, S. Zhuang, H. Sheng, G. Kang, and J. Xiao, "An ensemble prediction intervals approach for short-term PV power forecasting," *Sol. Energy*, vol. 155, pp. 1072–1083, Oct. 2017.

[107] H. Quan, D. Srinivasan, and A. Khosravi, "Uncertainty handling using neural network-based prediction intervals for electrical load forecasting," *Energy*, vol. 73, pp. 916–925, Aug. 2014.

[108] J. Wang, Y. Gao, and X. Chen, "A novel hybrid interval prediction approach based on modified lower upper bound estimation in combination with multi-objective salp swarm algorithm for short-term load forecasting," *Energies*, vol. 11, no. 6, p. 1561, Jun. 2018.

[109] M. Rana, I. Koprinska, A. Khosravi, and V. G. Agelidis, "Prediction intervals for electricity load forecasting using neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–8.

[110] R. Taormina and K.-W. Chau, "ANN-based interval forecasting of streamflow discharges using the LUBE method and MOFIPS," *Eng. Appl. Artif. Intell.*, vol. 45, pp. 429–440, Oct. 2015.

[111] C. Lian, Z. Zeng, W. Yao, H. Tang, and C. L. P. Chen, "Landslide displacement prediction with uncertainty based on neural networks with random hidden weights," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2683–2695, Dec. 2016.

[112] M. A. Hosen, A. Khosravi, S. Nahavandi, and D. Creighton, "Improving the quality of prediction intervals through optimal aggregation," *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4420–4429, Jul. 2015.

[113] M. A. Hosen, A. Khosravi, S. Nahavandi, and D. Creighton, "Prediction interval-based neural network modelling of polystyrene polymerization reactor—A new perspective of data-based modelling," *Chem. Eng. Res. Des.*, vol. 92, no. 11, pp. 2041–2051, Nov. 2014.

[114] X.-Y. Chen and K.-W. Chau, "Uncertainty analysis on hybrid double feedforward neural network model for sediment load estimation with LUBE method," *Water Resour. Manage.*, vol. 33, no. 10, pp. 3563–3577, Aug. 2019.

[115] H. Zhang, J. Zhou, L. Ye, X. Zeng, and Y. Chen, "Lower upper bound estimation method considering symmetry for construction of prediction intervals in flood forecasting," *Water Resour. Manage.*, vol. 29, no. 15, pp. 5505–5519, Dec. 2015.

[116] A. Kavousi-Fard, A. Khosravi, and S. Nahavandi, "Reactive power compensation in electric arc furnaces using prediction intervals," *IEEE Trans. Ind. Electron.*, vol. 64, no. 7, pp. 5295–5304, Jul. 2017.

[117] L. Ye, J. Zhou, H. V. Gupta, H. Zhang, X. Zeng, and L. Chen, "Efficient estimation of flood forecast prediction intervals via single- and multi-objective versions of the LUBE method," *Hydrolog. Processes*, vol. 30, no. 15, pp. 2703–2716, Jul. 2016.

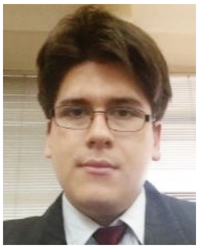
[118] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1613–1622.

[119] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Proc. Adv. neural Inf. Process. Syst.*, 2015, pp. 2575–2583.

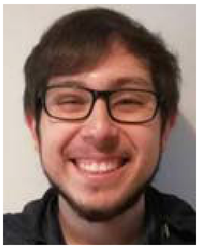
[120] I. Osband, J. Aslanides, and A. Cassirer, "Randomized prior functions for deep reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8617–8629.

[121] S. Chen, S. A. Billings, and P. M. Grant, "Non-linear system identification using neural networks," *Int. J. Control*, vol. 51, no. 6, pp. 1191–1214, Jan. 1990.

[122] A. Nespoli, M. Mussetta, E. Ogliari, S. Leva, L. Fernández-Ramírez, and P. García-Triviño, "Robust 24 hours ahead forecast in a microgrid: A real case study," *Electronics*, vol. 8, no. 12, p. 1434, Dec. 2019.



OSCAR CARTAGENA (Graduate Student Member, IEEE) was born in Santiago, Chile. He received the B.Sc. and M.Sc. degree in electrical engineering from the University of Chile, Santiago, Chile, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering. His current research interests include model predictive control, fuzzy systems, and prediction interval modeling.



SEBASTIÁN PARRA (Member, IEEE) was born in Santiago, Chile. He received the B.Sc. degree in electrical engineering from the University of Chile, Santiago, Chile, in 2018, where he is currently pursuing the M.Sc. degree in electrical engineering. His current research interests include image processing and computational intelligence-based predictive modeling.

DIEGO MUÑOZ-CARPINTERO (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the Universidad de Chile, in 2009 and 2010, respectively, and the D.Phil. degree in control engineering from the University of Oxford, in 2014. From 2015 to 2016, he was a Research Fellow at Nanyang Technological University. From 2017 to 2019, he was a Postdoctoral Researcher at the University of Chile. He is currently an Assistant Professor at the Institute of Engineering Science, Universidad de O'Higgins. His research interests include control theory in the areas of predictive, robust and stochastic control; intelligent control; fuzzy and neural modelling; decision making under uncertainty; and their application to energy efficient systems.

LUIS G. MARÍN (Member, IEEE) was born in Florencia-Caquetá, Colombia. He received the degree in electronic engineering from the Universidad Autónoma de Colombia, Bogotá, Colombia, in 2003, the M.Sc. degree in information sciences and communications from the Universidad Distrital Francisco José de Caldas, Bogotá, in 2012, and the Ph.D. degree in electrical engineering from the University of Chile, Santiago, Chile, in 2018. He is currently a Postdoctoral Assistant with the Faculty of Engineering, Andes University, Bogotá. His current research interests include predictive control, renewable energy plants, and control strategies for operation and coordination of microgrids.

DORIS SÁEZ (Senior Member, IEEE) was born in Panguipulli, Chile. She received the M.Sc. and Ph.D. degrees in electrical engineering from the Pontificia Universidad Católica de Chile, Santiago, Chile, in 1995 and 2000, respectively. She is currently a Full Professor with the Department of Electrical Engineering and the Head of the Indigenous People Program, Faculty of Mathematical and Physical Sciences, University of Chile, Santiago. She has coauthored the books *Hybrid Predictive Control for Dynamic Transport Problems* (Springer Verlag, 2013) and *Optimization of Industrial Processes at Supervisory Level: Application to Control of Thermal Power Plants* (Springer-Verlag, 2002). Her research interests include predictive control, fuzzy control design, fuzzy identification, and control of microgrids. She also serves as an Associate Editor for the IEEE Transactions on Smart Grid.