

Wind energy forecasting with missing values within a fully conditional specification framework

Abstract

Wind power forecasting is essential to power system operation and electricity markets. As abundant data became available thanks to the deployment of measurement infrastructures and the democratization of meteorological modelling, extensive data-driven approaches have been developed within both point and probabilistic forecasting frameworks. These models usually assume that the dataset at hand is complete and overlook missing value issues that often occur in practice. In contrast to that common approach, we rigorously consider here the wind power forecasting problem in the presence of missing values, by jointly accommodating imputation and forecasting tasks. Our approach allows inferring the joint distribution of input features and target variables at the model estimation stage based on incomplete observations only. We place emphasis on a fully conditional specification method owing to its desirable properties, e.g., being assumption-free when it comes to these joint distributions. Then, at the operational forecasting stage, with available features at hand, one can issue forecasts by implicitly imputing all missing entries. The approach is applicable to both point and probabilistic forecasting, while yielding competitive forecast quality within both simulation and real-world case studies. It confirms that by using a powerful universal imputation method based on fully conditional specification, the proposed universal imputation approach is superior to the common impute-then-predict approach, especially in the context of probabilistic forecasting.

Keywords: Wind power, Probabilistic forecasting, Missing values, Multiple imputation

1. Introduction

1.1. Background

As a cornerstone to achieve net-zero emissions in the energy sector, wind power has proliferated over recent decades. However, the stochastic nature of wind power generation challenges power system operation and electricity markets, which has therefore motivated wind power forecasting (WPF) research. WPF is usually classified into short-term forecasting (hours to few days) that takes numerical weather predictions as input features and very short-term forecasting (minutes to few hours) that utilizes recent observations as input features. Recently WPF has achieved several advances by employing cutting-edge statistical and machine learning approaches, e.g. deep learning (Goodfellow et al., 2016) and lightGBM (Ke et al., 2017), as well as the modeling of underlying stochastic processes through the in-

vestigation of its spatial-temporal dynamics (Cavalcante et al., 2017; Messner and Pinson, 2019).

Meanwhile, the interest of the WPF community has shifted from point forecasting to probabilistic forecasting; see recent review by Hong et al. (2020). Probabilistic wind power forecasting (PWPF) communicates the probability distribution of wind power generation at a future time based on gathered information up to the issue time, usually in the form of quantiles, prediction intervals, and densities. It has attracted increasing attention in the power industry, especially after the 2014 Global Energy Forecasting Competition (GEFCom 2014) (Hong et al., 2016). In general, two approaches, namely parametric and non-parametric have been proposed for PWPF. The parametric approach is based on a distributional assumption, such as Gaussian, Beta, etc., the shape parameters of which are determined through statistical learning. In contrast, the non-parametric approach is free of such an assumption. One of the most popular non-parametric approaches relies on quantile regression (QR) (Koenker and Hallock, 2001), which involves a pinball loss function to guide the learning of conditional quantile functions. It is therefore easy to employ QR in advanced statistical learning models (for instance gradient boosting machine (Landry et al., 2016) and extreme learning machine (Wan et al., 2016)) by using the pinball loss as loss function at the model estimation phase. Besides, with the aim to characterize the whole distribution in a distribution-free manner, methods that simultaneously estimates several quantiles (Sangnier et al., 2016) and directly estimates the distribution based on conditional normalizing flow(s) (Wen et al., 2022) have been proposed.

Although several works have contributed forecasting methods and products to the WPF community, most of them assume that the dataset at hand is complete and overlook the widespread missing value problems, due to sensor failures and communication errors for instance. Intuitively, missing value issues pose problems at both model estimation and operational forecasting stages, ultimately compromising forecast quality. Obviously, for models estimated through gradient-based optimization, the training datasets cannot contain missing values, otherwise the gradients of the parameters cannot be calculated at the model estimation stage. Therefore, rows of the learning set containing both missing values and observations are often deleted, even if the missing information is minimal. It means that valuable information is also discarded in the process of removing the missing values. In addition, even with estimated models at hand, missing value problems still affect operational forecasting, possibly obliging forecasters to revert to naive models such as climatology (i.e., long-term averages) as surrogates. Therefore, it remains an open issue to investigate the influence of missing values and develop WPF approaches that accommodate missing values.

1.2. Related works

An intuitive and popular approach to the problem (though, not used by the WPF community) is to impute these missing values before training models and issuing operational forecasts (Liu et al., 2018). It is referred to as “*impute, then predict*” (ITP) approach in this paper. For example, the classic forecasting package “forecast” (Hyndman and Khandakar, 2008) provides an option that uses linear interpolation to impute missing values. Obviously, a spectrum of imputation methods can be employed; see a thorough review by

Van Buuren (2018). Then, an associated question is how to choose the imputation method. The recent study by Tawn et al. (2020) suggests that the influence of imputation on model estimation and operational forecasting stages is ambiguous. Concretely, they concluded that advanced imputation methods are beneficial to model estimation. However, at the operational forecasting, it turns out that retraining models without missing features results in better performance. It is natural to consider the retraining approach, as it only uses actual observations to estimate parameters, and hence prevents using aforementioned imputation procedure. However, the learning then only relies on a subset of the data available, while the information potentially contained in the discarded part is lost. In addition, this approach may suffer the curse of dimensionality since having to train models for all combinations of input features. This may yield a substantial increase in computational costs.

In addition to the aforementioned approaches, several works have focused on adapting forecasting methods to be used in the presence of missing values. A classic approach is based on state-space modeling, where the Kalman filter is modified to allow accommodating incomplete observations. For example, autoregressive moving average models (Jones, 1980) and autoregressive integrated moving average models (Kohn and Ansley, 1986) have been represented in state-space form and adapted to tackle missing value problems. Although these works have shed light on forecasting in the presence of missing values, they are only applicable to point forecasting and restricted to linear models. Recent advanced models such as GRU-D (Che et al., 2018) and BRITS (Cao et al., 2018) have been proposed based on the long-short term memory model (Hochreiter and Schmidhuber, 1997), by using the intermediate results (which can be also interpreted as latent states) of the neural network model to impute missing values. This idea has been successfully applied in the recent popular package DeepAR (Salinas et al., 2020). However, they still require to impute missing values via the recurrent neural network structure before performing the forecasting task.

1.3. Proposed method and contributions

There is no such a clearly defined boundary between imputation and forecasting, as explained by Golyandina and Osipov (2007). Indeed, a forecasting problem can be considered as an imputation problem in the situation where missing values are systematically located at the end of a sequence. Furthermore, both imputation and forecasting tasks assume the continuation of the underlying structure within the data, and consequently leverage observations to predict unknown values. That is, it is feasible to develop a model that can infer the structure based on observations and seamlessly perform the imputation and forecasting tasks, which is referred to as “*universal imputation*” (UI) approach in this paper. As a result, in what follows, we may interchangeably use the terms “impute” and “forecast”. Particularly, it is assumed that observations are missing at random, which means missingness patterns are independent of the missing values. The distribution of missingness can then be left aside when inferring the underlying structure of interest. However, it does not mean that this concept of missingness at random is restricted to the case of data missing in a pointwise and sporadic fashion. Even in the case of block missingness (i.e., data missing over time intervals), as long as the data is missing at random (hence, independently of the values for the process of interest or relevant exogenous processes), our approach can be

employed. As a consequence, the problem boils down to estimating the parameters of a model based on incomplete observations only. With this idea in mind, [You et al. \(2020\)](#) considered the point forecasting problem and proposed to model the correlation structure between input features and targets via a graph neural network, where imputation of missing features and prediction of targets can be simultaneously performed. In contrast here, we place ourselves within a probabilistic setting directly, for which it is then also possible to derive point forecasts. Focus is on a very-short-term wind power forecasting applications, where missing value issues often occur, though the method is generic and could be then used by others for different applications where challenges accommodating missing values are also present.

Unlike the usual probabilistic forecasting approaches that model conditional probability distributions (for the target variable) directly, in this work we model the joint multivariate probability distribution of input features and targets. As discussed by [Stone \(1991\)](#), with the estimated multivariate probability distribution at hand, one can obtain conditional distributions via marginalization, although it is computationally inefficient compared to the usual conditional probability distribution modeling. This approach is appealing in the presence of missing values. That is, with the estimated multivariate probability distribution, one can marginalize over missing variables to obtain probabilistic forecasts. Then, the goal at the model estimation stage is to estimate the parameters of such a distribution based on incomplete observations. At the operational forecasting stage, targets to be predicted are treated as missing values, and imputed via the estimated distribution. In this work, we implement this idea based on the multiple imputation method ([Dempster et al., 1977](#)), which allows to impute missing values with several equally likely realizations from the distribution and thus provides probabilistic forecasts for the targets. Instead of assuming a special family of distributions and inferring its parameters, we adopt the fully conditional specification (FCS) approach ([Van Buuren et al., 2006](#)), which implicitly specifies the multivariate distribution as a collection of conditional distributions on a variable-by-variable basis. At the model estimation stage, parameters for each conditional distribution are iteratively estimated through a Gibbs sampling procedure. At the operational forecasting stage, missing values are also iteratively imputed on a variable-by-variable basis.

The proposed method is validated based on a simulation study and real-world case studies with wind power data from the USA. The simulation study is based on synthetic data (for both AR and VAR processes) and Monte-Carlo simulations, to illustrate and underline the salient features of our approach. It also allows analysing the impact of certain characteristics e.g. rate of missingness on the performance of the approach, while remaining in a controlled environment within which the results are due to changes in the design, and not to some spurious effects often observed with real-world data. Real-world data from the US are used for benchmarking instead, and to investigate various aspects of the applicability of the approach in an operational context, e.g., with focus on pointwise vs. block missingness, various rates of missingness, univariate and multivariate setups, etc. The results show that the proposed approach is superior to existing ITP approaches. The main contributions of this paper are two-fold. One of them is the proposal of a universal imputation approach, which is general, though inspired by the problem of wind power forecasting in the presence

of missing values. Such a universal imputation approach jointly accommodates imputation and forecasting tasks within the universal multiple imputation framework. By design, this approach allows to generate both point and probabilistic forecasts. The other contribution is to show its applicability to wind power forecasting with missing values, where different types and rates of missingness are present.

The remaining parts of this paper are organized as follows. Section 2 formulates the problem, whereas Section 3 describes the proposed approach for forecasting in the presence of missing values. Next, the simulation study to show the applicability of the proposed approach is elaborated in Section 4. Section 5 presents case studies with results and discussion. Section 6 concludes the paper.

Notations: In general, we use uppercase letters to denote random variables and lowercase letters to denote the realizations of these random variables. For instance, Y_1 denotes a random variable and y_1 its realization. A collection of random variables are represented as a tuple, which is bracketed with parentheses, such as (Y_1, Y_2) and (Y_1, \dots, Y_{10}) . Boldface lowercase and uppercase letters respectively indicate vectors and matrices. Particularly, we use row and column slices to represent parts of a matrix. For instance, let \mathbf{Z} represent a matrix, \mathcal{I} and \mathcal{J} denote row indices and column indices. Then, $\mathbf{Z}[\mathcal{I}; \mathcal{J}]$ represents a part of matrix \mathbf{Z} indexed by \mathcal{I} and \mathcal{J} . And, $(\cdot)^\top$ denotes the transpose of matrices. A time series is represented as $\{y_t, t = 1, 2, \dots\}$ indexed by time t , which is a realization of a stochastic process $\{Y_t, t = 1, 2, \dots\}$. We also write them as $\{y_t\}$ and $\{Y_t\}$ for short.

2. Preliminaries

We first describe the framework for very-short term wind power forecasting, for both point and probabilistic forecasting cases. Subsequently, we detail the challenges induced by missing values at both model estimation and operational forecasting stages.

2.1. Problem Formulation

Assume we have p wind farms in a region that can share information to improve forecasting accuracy as suggested by [Cavalcante et al. \(2017\)](#). When p equals to 1, it reduces to the common single wind farm case. At wind farm n , let $y_{n,t} \in [0, P_n]$ (where P_n is its capacity) denote the wind power generation value at time t , which is a realization of the random variable $Y_{n,t}$. Let $\Omega_{n,t}$ denote the information tuple of wind farm n up to time t , which would contain values over previous time steps and possibly other relevant information such as weather observations and numerical weather forecasts. And let Ω_t represent the tuple that contains information of all sites up to time t , i.e., $\Omega_t = (\Omega_{1,t}, \dots, \Omega_{p,t})$. Generally, the aim is to issue forecasts with lead time h , i.e., the characteristics of $Y_{1,t+1}, \dots, Y_{1,t+h}, \dots, Y_{p,t+1}, \dots, Y_{p,t+h}$, given information Ω_t . The forecasting task can be decoupled into several sub-problems, each of which focuses on a specific site and time, for instance forecasting the characteristics of $Y_{n,t+h}$ based on the whole information pool Ω_t . Then the point forecast for $Y_{n,t+h}$ given by a model \mathcal{M} with parameters $\hat{\Theta}_t$ is usually defined as

$$\hat{y}_{n,t+h|t} = \mathbb{E}[Y_{n,t+h} | \mathcal{M}, \hat{\Theta}_t, \Omega_t], \quad (1)$$

where $\mathbb{E}[\cdot]$ denotes the expectation of random variables, and $\hat{\Theta}_t$ changes with time t . In this paper, let us assume the stochastic process $\{Y_{1,t}, Y_{2,t}, \dots, Y_{p,t}\}$ is stationary. Then, the density function $f_{Y_{1,t}, \dots, Y_{p,t+h}}$ is invariant for changes in time (De Gooijer et al., 2017), which means parameters $\hat{\Theta}_t$ do not vary with time and are denoted as $\hat{\Theta}$. Then, one can estimate the parameters based on collected data via statistical learning methods. We rewrite (1) as

$$\hat{y}_{n,t+h|t} = \mathbb{E}[Y_{n,t+h} | \mathcal{M}, \hat{\Theta}, \Omega_t]. \quad (2)$$

The probabilistic forecast for time $t+h$ given by \mathcal{M} is communicated as a density function, i.e.,

$$\hat{f}_{n,t+h|t}(y) = f_{Y_{n,t+h}}(y | \mathcal{M}, \hat{\Theta}, \Omega_t). \quad (3)$$

Indeed, with the estimated density function at hand, one can easily obtain point forecast via:

$$\hat{y}_{n,t+h|t} = \int_y y \hat{f}_{n,t+h|t}(y) dy. \quad (4)$$

For simplicity of notations, let us focus on the predictive density $\hat{f}_{n,t+h|t}$ given the information set Ω_t at time t . We denote the input features as \mathbf{x}_t and the realization of target $Y_{n,t+h}$ as y_t . The information one has access to N sample pairs $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ serves as a basis for training. They can be written in the form of a matrix, i.e., $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top$ as well as $\mathbf{Y} = [y_1, \dots, y_N]^\top$. The matrix \mathbf{X} has dimensions $N \times pk$, whereas \mathbf{Y} is a vector with N elements. Now the density forecast described in (3) boils down to conditional probability density function estimation, which is performed via statistical learning. In very-short term WPF, one commonly uses past wind power generation values of length k as input features, i.e., a vector $[y_{n,t-k+1}, \dots, y_{n,t}]^\top \in [0, P_n]^k$ for the n^{th} site. Therefore, considering all sites together, the vector of input features is given by

$$\mathbf{x}_t = [y_{1,t-k+1}, \dots, y_{1,t}, \dots, y_{p,t-k+1}, \dots, y_{p,t}]^\top \in [0, P_1]^k \times [0, P_2]^k \times \dots \times [0, P_p]^k.$$

Obviously, features in \mathbf{x}_t have some form of dependency, which breaks down the i.i.d assumption in statistical learning. However, it is still common to place oneself in a regression framework for estimation and overlook this dependency issue, as for instance done recently also for global/local model estimation (Montero-Manso and Hyndman, 2021) and estimation in deep learning models (Benidis et al., 2022).

Based on a stationarity assumption, the sample pairs $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ can be regarded as identically distributed. For simplicity, we introduce two random variables X and Y for these samples. It allows us to model the joint distribution $f_{X,Y}(\mathbf{x}, y)$ via \mathcal{M} with estimated parameters $\hat{\Theta}$, i.e., $f_{X,Y}(\mathbf{x}, y; \mathcal{M}, \hat{\Theta})$, and derive $f_{Y|X}(y|\mathbf{x})$ via the conditional probability formula. It is described as

$$f_{Y|X}(y|\mathbf{x}; \mathcal{M}, \hat{\Theta}) = \frac{f_{X,Y}(\mathbf{x}, y; \mathcal{M}, \hat{\Theta})}{f_X(\mathbf{x}; \mathcal{M}, \hat{\Theta})} = \frac{f_{X,Y}(\mathbf{x}, y; \mathcal{M}, \hat{\Theta})}{\int_y f_{X,Y}(\mathbf{x}, y; \mathcal{M}, \hat{\Theta}) dy}. \quad (5)$$

With the estimated joint distribution $f_{X,Y}(\mathbf{x}, y; \mathcal{M}, \hat{\Theta})$ at hand, at any time t , given contextual information \mathbf{x}_t , one can issue the forecast $\hat{f}_{Y|X}(y_t|\mathbf{x}_t; \mathcal{M}, \hat{\Theta})$ via (5). In this paper,

\mathcal{M} is set as an imputation model and implicitly defined by a collection of conditional distributions. Each conditional distribution is implemented by predictive mean matching that relies on a function, for instance g_j parameterized by $\hat{\theta}_j$. As we are considering the joint distribution now, we can concatenate \mathbf{x}_t and y_t as \mathbf{z}_t , i.e., $\mathbf{z}_t = [\mathbf{x}_t^\top, y_t]^\top$. Accordingly, the dataset is concatenated as the matrix \mathbf{Z} of shape $N \times (pk + 1)$, i.e.,

$$\mathbf{Z} = \begin{bmatrix} \mathbf{x}_1^\top & y_1 \\ \mathbf{x}_2^\top & y_2 \\ \vdots & \vdots \\ \mathbf{x}_N^\top & y_N \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1^\top \\ \mathbf{z}_2^\top \\ \vdots \\ \mathbf{z}_N^\top \end{bmatrix}.$$

We refer to the i -th row, j -th column, and (i, j) -th entry of \mathbf{Z} as \mathbf{z}_i , \mathbf{Z}_j , and $z_{i,j}$ respectively. And we introduce a random variable $Z = (X, Y)$ that concatenates X and Y , which contains $pk + 1$ variables (recall that X has pk variables, as it represents information from p sites), i.e., $Z = (Z_1, Z_2, \dots, Z_{pk+1})$. Then, the distribution of Z is modeled by $f_Z(\mathbf{z}; \mathcal{M}, \hat{\Theta})$. In particular, let Z_{-j} denote the collection of random variables in Z except Z_j , i.e., $Z_{-j} = (Z_1, \dots, Z_{j-1}, Z_{j+1}, \dots, Z_{pk+1})$. Accordingly, let \mathbf{z}_{-j} denote the realization of Z_{-j} .

We assume values are missing at random. This is to be understood in a way that is more general than data missing sporadically and at random times. More formally, missingness at random means that the fact a data entry is missing or not is independent of the process itself, or of some exogenous process. For the wind power application, data missing not at random could be for the case there are systematic sensor failures for power generation values below a given threshold, or systematic communication failures when wind comes for a given direction. In addition, missingness at random is not restricted to the case data is missing at single times. It can also be for the case of data missing over time intervals (i.e., block missingness). This assumption of missingness at random is expected to be sound for wind power applications, though this should be confirmed on a case-by-case basis based on advanced data analysis.

Missing missing values are likely to occur in every element of \mathbf{z}_t . Let us introduce a vector \mathbf{m}_t to indicate the missingness of \mathbf{z}_t . Concretely, $m_{t,j} = 1$ indicates that $z_{t,j}$ is missing, whereas $m_{t,j} = 0$ indicates that $z_{t,j}$ is observed. Accordingly, the matrix \mathbf{M} indicates the missingness of \mathbf{Z} . Let $\mathcal{J}_{\mathbf{z}_t, \mathbf{M}}$ denote the indices of missingness of \mathbf{z}_t , i.e., $\mathcal{J}_{\mathbf{z}_t, \mathbf{M}} = \{j \mid m_{t,j} = 1\}$, and $\mathcal{J}_{\mathbf{z}_t, \mathbf{O}}$ denote the indices of observations, i.e., $\mathcal{J}_{\mathbf{z}_t, \mathbf{O}} = \{j \mid m_{t,j} = 0\}$. Therefore, the observed and missing parts of \mathbf{z}_t are represented by $\mathbf{z}_t[\mathcal{J}_{\mathbf{z}_t, \mathbf{O}}]$ and $\mathbf{z}_t[\mathcal{J}_{\mathbf{z}_t, \mathbf{M}}]$, which are written as \mathbf{z}_t^{obs} and \mathbf{z}_t^{mis} for simplicity. The corresponding random variables for \mathbf{z}_t^{obs} and \mathbf{z}_t^{mis} are denoted as Z^{obs} and Z^{mis} . When y_t is missing, $\mathbf{z}_t^{mis} = [\mathbf{x}_t^{mis^\top}, y_t]^\top$ where \mathbf{x}_t^{mis} is the missing part of \mathbf{x}_t . The corresponding random variables for \mathbf{x}_t^{mis} are denoted as X^{mis} . For example, Figure 1 presents the matrix $\mathbf{Z} = [z_{i,j}]_{4 \times 4}$, where blue blocks indicate observations and yellow blocks indicate missing values. As shown, the first row of \mathbf{Z} is denoted as \mathbf{z}_1 , the second entry of which, i.e., $z_{1,2}$ is missing. Then, the indices of missing values and observations of \mathbf{z}_1 are $\mathcal{J}_{\mathbf{z}_1, \mathbf{M}} = \{2\}$ and $\mathcal{J}_{\mathbf{z}_1, \mathbf{O}} = \{1, 3, 4\}$. Accordingly, we have $\mathbf{z}_1^{obs} = [z_{1,1}, z_{1,3}, z_{1,4}]^\top$, $\mathbf{z}_1^{mis} = [z_{1,2}]$. The corresponding random variables for \mathbf{z}_1^{obs} and \mathbf{z}_1^{mis} are denoted as $Z^{obs} = (Z_1, Z_3, Z_4)$ and $Z^{mis} = Z_2$. Also, let $\mathcal{I}_{\mathbf{z}_j, \mathbf{M}}$ denote the

indices of missing values in \mathbf{Z}_j , i.e., $\mathcal{I}_{\mathbf{Z}_j, M} = \{i \mid m_{i,j} = 1\}$, and $\mathcal{I}_{\mathbf{Z}_j, O}$ denote the indices of observations in \mathbf{Z}_j , i.e., $\mathcal{I}_{\mathbf{Z}_j, O} = \{i \mid m_{i,j} = 0\}$. Then the missing and observed parts of \mathbf{Z}_j are $\mathbf{Z}[\mathcal{I}_{\mathbf{Z}_j, M}; j]$ and $\mathbf{Z}[\mathcal{I}_{\mathbf{Z}_j, O}; j]$, which are respectively written as \mathbf{Z}_j^{mis} and \mathbf{Z}_j^{obs} for simplicity. In Figure 1, \mathbf{Z}_1 represents the first column of \mathbf{Z} , the second entry of which is missing. Accordingly, we have $\mathcal{I}_{\mathbf{Z}_1, M} = \{2\}$, $\mathcal{I}_{\mathbf{Z}_1, O} = \{1, 3, 4\}$, $\mathbf{Z}_1^{obs} = [z_{1,1}, z_{3,1}, z_{4,1}]^\top$, and $\mathbf{Z}_1^{mis} = [z_{2,1}]$.

$z_{1,1}$	$z_{1,2}$	$z_{1,3}$	$z_{1,4}$
$z_{2,1}$	$z_{2,2}$	$z_{2,3}$	$z_{2,4}$
$z_{3,1}$	$z_{3,2}$	$z_{3,3}$	$z_{3,4}$
$z_{4,1}$	$z_{4,2}$	$z_{4,3}$	$z_{4,4}$

Figure 1: Illustration of a dataset \mathbf{Z} . Here we take $p = 1, k = 3, h = 1$ as an example. Blue blocks indicate observations, whereas yellow blocks indicate missing values.

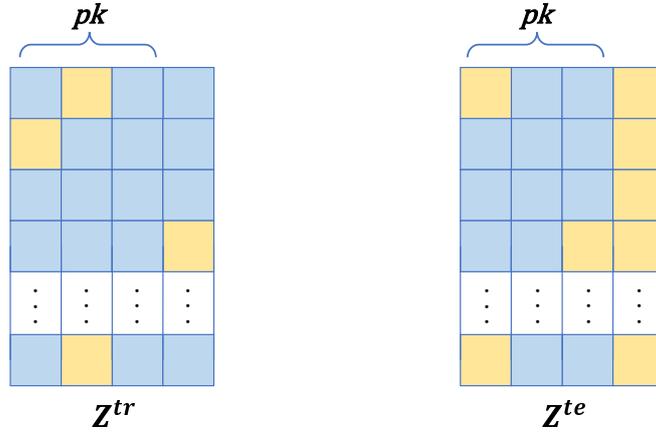


Figure 2: Illustration of training and test datasets. Here we take $p = 1, k = 3, h = 1$ as an example. Blue blocks indicate observations, whereas yellow blocks indicate missing values.

Therefore, at the model estimation phase, we concatenate features and targets to form a training dataset \mathbf{Z}^{tr} with some missing values, based on which an imputation model \mathcal{M} is trained. At the operational forecasting phase, the input vector \mathbf{x}_t at time t is available, part of which may be missing, and we focus on target y_t . Together, they form $\mathbf{z}_t = [\mathbf{x}_t^\top, y_t]^\top$. Then \mathbf{z}_t is imputed via the estimated model. For illustration, we present the training and test datasets for the single wind farm case in Figure 2. In the training dataset, missingness occurs in both input features and targets. In the test dataset, all targets are systematically missing.

2.2. Challenge at the model estimation stage

Usually, the learning process of parameters in density estimation problems is based on maximum likelihood, which involves the computation of likelihood. However, in the presence of missing values, the likelihood is blended with missingness indicators. With the assumption that values are missing at random, the parameters of underlying distributions can be estimated based on observations only. Let Θ denote the true parameters of \mathcal{M} . Consider the likelihood of a sample \mathbf{z}_t . It is described as

$$f_Z(\mathbf{z}_t, \mathbf{m}_t; \mathcal{M}, \Theta) = f_Z(\mathbf{z}_t^{obs}, \mathbf{z}_t^{mis}, \mathbf{m}_t; \mathcal{M}, \Theta), \quad (6)$$

where \mathbf{z}_t^{mis} is missing. The likelihood function can be marginalized with respect to \mathbf{z}_t^{mis} , i.e.,

$$\begin{aligned} f_{Z^{obs}}(\mathbf{z}_t^{obs}; \mathcal{M}, \Theta) &= \int f_{Z^{obs}, Z^{mis}}(\mathbf{z}_t^{obs}, \mathbf{z}_t^{mis}, \mathbf{m}_t; \mathcal{M}, \Theta) d\mathbf{z}_t^{mis} \\ &= \int f_{Z^{obs}, Z^{mis}}(\mathbf{z}_t^{obs}, \mathbf{z}_t^{mis}; \mathcal{M}, \Theta) d\mathbf{z}_t^{mis}. \end{aligned} \quad (7)$$

Therefore, to learn the parameters Θ , it is required to maximize the likelihood of observations only, i.e., $f_{Z^{obs}}(\mathbf{z}_t^{obs}; \mathcal{M}, \Theta)$. The estimate of Θ is denoted as $\hat{\Theta}$.

2.3. Challenge at the operational forecasting stage

In this section we assume that we already have distribution $f_Z(\mathbf{z}; \mathcal{M}, \hat{\Theta})$ with estimated parameters $\hat{\Theta}$ at hand, and show how to issue forecasts at the operational forecasting stage. If \mathbf{x}_t is fully observed, then \mathbf{x}_t is the observed part of \mathbf{z}_t , i.e., $\mathbf{z}_t^{obs} = \mathbf{x}_t$, whereas the missing part of \mathbf{z}_t is y_t . The forecast for \mathbf{x}_t can be expressed as

$$f_{Y|X}(y_t | \mathbf{x}_t; \mathcal{M}, \hat{\Theta}) = f_{Z^{mis}|Z^{obs}}(\mathbf{z}_t^{mis} | \mathbf{z}_t^{obs}; \mathcal{M}, \hat{\Theta}) = \frac{f_Z(\mathbf{z}_t^{obs}, \mathbf{z}_t^{mis}; \mathcal{M}, \hat{\Theta})}{\int_{\mathbf{z}_t^{mis}} f_Z(\mathbf{z}_t^{obs}, \mathbf{z}_t^{mis}; \mathcal{M}, \hat{\Theta}) d\mathbf{z}_t^{mis}}. \quad (8)$$

In the presence of missing values, the forecasting task is to issue $f_{Y|Z^{obs}}(y_t | \mathbf{z}_t^{obs})$ by utilizing the distribution $f_Z(\mathbf{z}; \mathcal{M}, \hat{\Theta})$. Indeed, \mathbf{z}_t^{mis} can be decomposed into \mathbf{x}_t^{mis} and y_t , i.e.,

$$f_{Z^{mis}|Z^{obs}}(\mathbf{z}_t^{mis} | \mathbf{z}_t^{obs}; \mathcal{M}, \hat{\Theta}) = f_{Y, X^{mis}|Z^{obs}}(y_t, \mathbf{x}_t^{mis} | \mathbf{z}_t^{obs}; \mathcal{M}, \hat{\Theta}). \quad (9)$$

Then the desired $f_{Y|Z^{obs}}(y_t | \mathbf{z}_t^{obs}; \mathcal{M}, \hat{\Theta})$ is derived by marginalizing $f_{Z^{mis}|Z^{obs}}(\mathbf{z}_t^{mis} | \mathbf{z}_t^{obs}; \mathcal{M}, \hat{\Theta})$ with respect to \mathbf{x}_t^{mis} , i.e.,

$$f_{Y|Z^{obs}}(y_t | \mathbf{z}_t^{obs}; \mathcal{M}, \hat{\Theta}) = \int f_{Y, X^{mis}|Z^{obs}}(y_t, \mathbf{x}_t^{mis} | \mathbf{z}_t^{obs}; \mathcal{M}, \hat{\Theta}) d\mathbf{x}_t^{mis}. \quad (10)$$

3. Forecasting with missing values via FCS

In this section, we develop a forecasting approach based on the proposed universal imputation strategy. For that, we employ the fully conditional specification approach, which in practice will be based on Gibbs sampling. It is described in the first part of the Section. This FCS approach requires a method to derive conditional distributions, which we describe in the second part. Eventually, it also relies on the choice for a regression model (random forests here), covered in the third part of the section. Finally, we will describe how the overall approach can be readily used for genuine forecasting with missing data.

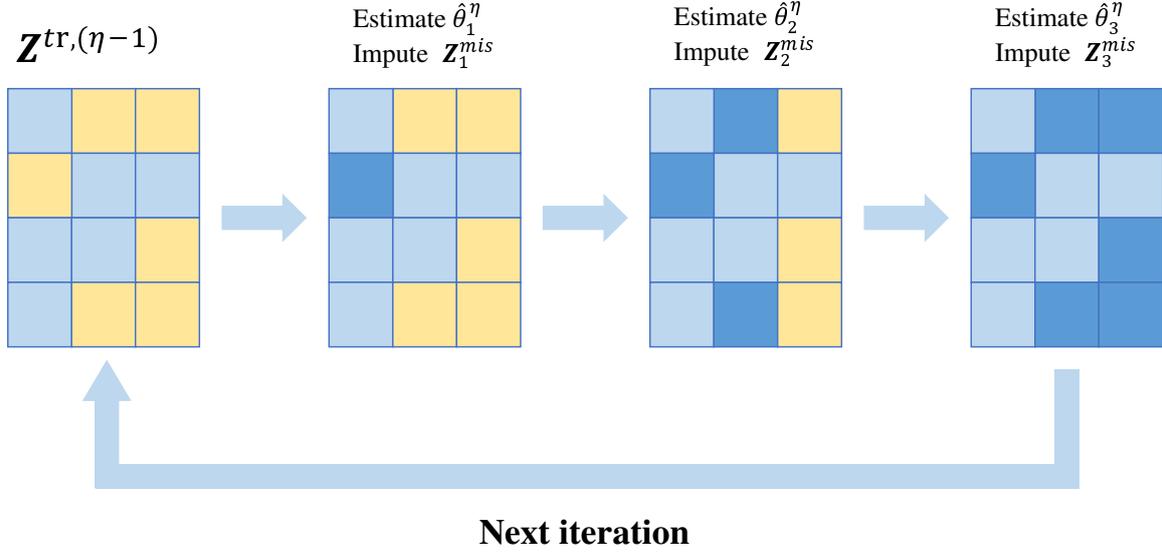


Figure 3: Illustration of the η -th iteration at the training stage. Light blue blocks indicate observations, yellow blocks indicate missing values, and dark blue blocks indicate imputation.

3.1. Fully conditional specification method

Instead of defining a multivariate distribution $f_{\mathbf{Z}}(\mathbf{z}; \mathcal{M}, \hat{\Theta})$ by assuming a specific distribution family, the FCS specifies a separate conditional distribution for each Z_j , just like a Gibbs sampler. Concretely, the conditional distribution for Z_j is modeled by g_j with parameters $\hat{\theta}_j$, and is denoted as $f_{Z_j|Z_{-j}}(z_j|\mathbf{z}_{-j}; g_j, \hat{\theta}_j)$. Therefore, the model \mathcal{M} is implemented via a bunch of models $\{g_j\}$, whereas $\hat{\Theta}$ is composed of all parameters $\{\hat{\theta}_j\}$. These parameters are estimated at the model estimation phase based on the training dataset \mathbf{Z}^{tr} . For simplicity of notations, we still use \mathbf{Z} in what follows to show how to estimate the parameters. Intuitively, before estimating $\hat{\theta}_j$, one needs to impute the missing values of \mathbf{Z}_{-j} . Then, parameters are estimated based on the imputed \mathbf{Z}_{-j} and \mathbf{Z}_j^{obs} . With the estimated conditional distribution $f_{Z_j|Z_{-j}}(z_j|\mathbf{z}_{-j}; g_j, \hat{\theta}_j)$, one can impute \mathbf{Z}_j^{mis} based on the corresponding conditionals in \mathbf{Z}_{-j} . That is, both the estimation of $\hat{\theta}_j$ and the imputation of \mathbf{Z}_j^{mis} are based on the imputed \mathbf{Z}_{-j} . Obviously, the imputation of any column of \mathbf{Z}_{-j} , for instance \mathbf{Z}_q , relies on its conditional distribution $f_{Z_q|Z_{-q}}(z_q|\mathbf{z}_{-q}; g_q, \hat{\theta}_q)$, which requires \mathbf{Z}_j^{mis} to be imputed. In other words, the

estimation of $\hat{\theta}_j$ and the imputation of \mathbf{Z}_{-j} are coupled with each other. If one performs the parameter estimation and imputation sequentially for $j = 1, 2, \dots, pk + 1$, the estimation of $\hat{\theta}_j$ can only use initial imputation of $\mathbf{Z}_{j+1}^{mis}, \dots, \mathbf{Z}_{pk+1}^{mis}$. The updated imputation of $\mathbf{Z}_{j+1}^{mis}, \dots, \mathbf{Z}_{pk+1}^{mis}$ given by their estimated conditional distributions cannot be used for the estimation of $\hat{\theta}_j$. Therefore, we perform the imputation of \mathbf{Z}_j and the estimation of $\hat{\theta}_j$ in an iterative manner. Then, at next iteration, the updated imputation of $\mathbf{Z}_{j+1}^{mis}, \dots, \mathbf{Z}_{pk+1}^{mis}$ can be used for the estimation of $\hat{\theta}_j$. For example, denote the estimated parameters $\hat{\theta}_j$ at the η -th iteration as $\hat{\theta}_j^{(\eta)}$, and the imputed complete column as $\mathbf{Z}_j^{(\eta)}$. At the $\eta + 1$ -th iteration, $\mathbf{Z}_{j+1}^{(\eta)}, \dots, \mathbf{Z}_{pk+1}^{(\eta)}$ can be used for the estimation of $\hat{\theta}_j^{(\eta+1)}$. Before the iterative estimation, all missing values are initially imputed as 0; therefore each column \mathbf{Z}_j becomes complete and is written as $\mathbf{Z}_j^{(0)}$. After all iterations, the ultimate estimation for θ_j is denoted as $\hat{\theta}_j$. Here, we set the stopping criterion as the round of iteration, as suggested by (Van Buuren et al., 2006). The caveat is that FCS method cannot guarantee the existence of joint distribution. Luckily, it is a relatively minor problem in practice, especially when missing rate is modest. We illustrate the steps of the η -th iteration in Figure 3.

Concretely, at the η -th iteration, before estimating $\hat{\theta}_j^{(\eta)}$, we have $\mathbf{Z}_1^{(\eta)}, \dots, \mathbf{Z}_{j-1}^{(\eta)}, \mathbf{Z}_{j+1}^{(\eta-1)}, \dots, \mathbf{Z}_{pk+1}^{(\eta-1)}$ at hand, which are written compactly as $\mathbf{Z}_{-j}^{(\eta)}$ in the form of a matrix, i.e.,

$$\mathbf{Z}_{-j}^{(\eta)} = [\mathbf{Z}_1^{(\eta)}, \dots, \mathbf{Z}_{j-1}^{(\eta)}, \mathbf{Z}_{j+1}^{(\eta-1)}, \dots, \mathbf{Z}_{pk+1}^{(\eta-1)}]. \quad (11)$$

Then $\hat{\theta}_j^{(\eta)}$ is estimated based on $\mathbf{Z}_{-j}^{(\eta)}$ and \mathbf{Z}_j^{obs} via maximum likelihood:

$$\hat{\theta}_j^{(\eta)} = \arg \max_{\theta_j} \sum_{i \in \mathcal{I}_{j,obs}} \log f_{Z_j|Z_{-j}}(z_{i,j} | \mathbf{z}_{i,-j}; g_j, \theta_j). \quad (12)$$

Thus we derive the estimated conditional distribution $f_{Z_j|Z_{-j}}(z_j | \mathbf{z}_{-j}; g_j, \hat{\theta}_j^{(\eta)})$, based on which we can impute \mathbf{Z}_j^{mis} . For instance, to impute the value $z_{i,j}$ in \mathbf{Z}_j^{mis} , we sample from $f_{Z_j|Z_{-j}}(z_j | \mathbf{z}_{i,-j}; g_j, \hat{\theta}_j^{(\eta)})$, which is described as:

$$z_{i,j}^{(\eta)} \sim f_{Z_j|Z_{-j}}(z_j | \mathbf{z}_{i,-j}; g_j, \hat{\theta}_j^{(\eta)}), \quad i \in \mathcal{I}_{j,mis}. \quad (13)$$

As \mathbf{Z}_j^{obs} is observed, we do not change the values, i.e.,

$$z_{i,j}^{(\eta)} = z_{i,j}^{(\eta-1)}, \quad i \in \mathcal{I}_{j,obs}. \quad (14)$$

Then we write all $z_{i,j}^{(\eta)}$ in the form of a vector, which is denoted as $\mathbf{Z}_j^{(\eta)}$ i.e.,

$$\mathbf{Z}_j^{(\eta)} = [z_{1,j}^{(\eta)}, \dots, z_{N,j}^{(\eta)}]^\top. \quad (15)$$

This procedure goes sequentially for $j = 1, \dots, pk + 1$. We note that the method can be executed multiple times in parallel to obtain multiple imputations. Besides, the model g_j for $f_{Z_j|Z_{-j}}(z_j | \mathbf{z}_{-j}; g_j, \hat{\theta}_j)$ needs to be specified, which is described in next section.

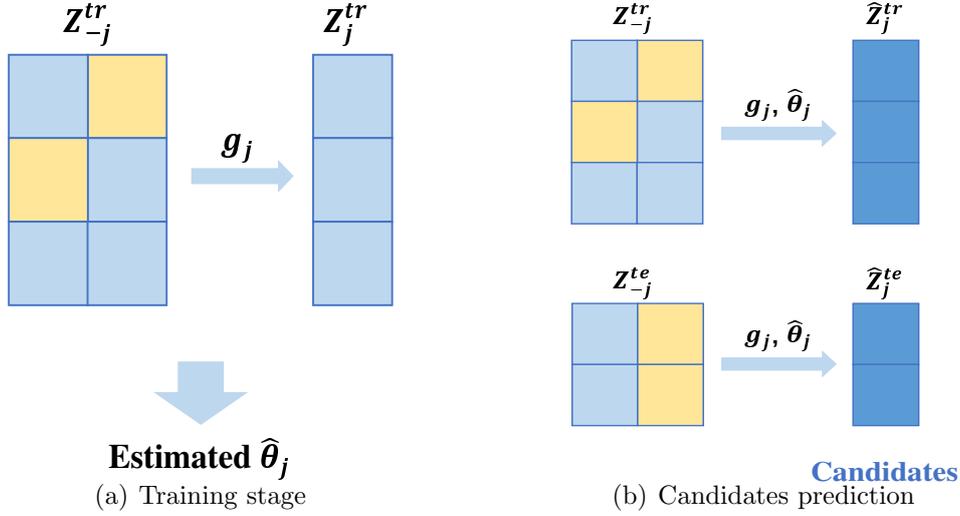


Figure 4: Illustration of key components of predictive mean matching.

3.2. Predictive mean matching

In this paper, $f_{Z_j|Z_{-j}}(z_j|z_{-j}; \hat{\theta}_j, g_j)$ is specified based on the predictive mean matching (Little and Rubin, 2019), which is free of distributional assumptions. Specifically, here g_j is not a real distribution model, but specified as a regression model. The distribution is given by a sampling procedure based on g_j . For each missing entry, we form a set of candidates from complete cases whose predicted values are close to the predicted value for the missing entry. Now we use parameters θ_j to specify the regression model g_j that maps z_{-j} to z_j , i.e.,

$$z_j = g_j(z_{-j}; \theta_j) + \epsilon_j, \quad (16)$$

where ϵ_j represents noise. We illustrate the key operations of this method in Figure 4, i.e., the training of the regression model and the prediction of candidates. That is, we estimate parameters $\hat{\theta}_j$ based on training datasets Z_{-j}^{tr} and Z_j^{tr} . With the estimated model, we respectively predict targets for Z_{-j}^{tr} and Z_{-j}^{te} , which are called candidates and written as \hat{Z}_j^{tr} and \hat{Z}_j^{te} . Then, for each entry of \hat{Z}_j^{te} , we form a set of its d closest candidates in \hat{Z}_j^{tr} , from which we perform random sampling to obtain imputations.

At the η -th iteration of FCS, the regression model is trained based on $Z_{-j}^{(\eta)}[Z_{j,obs}, :]$ and Z_j^{obs} by minimizing the loss, i.e.,

$$\hat{\theta}_j^{(\eta)} = \arg \min_{\theta_j} \sum_{i \in \mathcal{I}_{j,obs}} \ell(z_{i,j} - g_j(z_{i,-j}^{(\eta)}; \theta_j)), \quad (17)$$

where $\ell(\cdot)$ is the mean squared error function. Then, we predict a candidate value for each $z_{i,-j}^{(\eta)}$ via the trained regression model, which is denoted as $\hat{z}_{i,j}^{(\eta)}$, i.e.,

$$\hat{z}_{i,j}^{(\eta)} = g_j(z_{i,-j}^{(\eta)}; \hat{\theta}_j^{(\eta)}). \quad (18)$$

Together, they are written in the form of a vector as $\hat{\mathbf{Z}}_j^{(\eta)}$, which is expressed as

$$\hat{\mathbf{Z}}_j^{(\eta)} = [\hat{z}_{1,j}^{(\eta)}, \hat{z}_{2,j}^{(\eta)}, \dots, \hat{z}_{N,j}^{(\eta)}]^\top. \quad (19)$$

To impute \mathbf{Z}_j^{mis} , let us focus on each missing entry of it, for instance $z_{i_m,j}$, $i_m \in \mathcal{I}_{j,mis}$, whose candidate is $\hat{z}_{i_m,j}^{(\eta)}$. Then we find d nearest candidates from $\hat{\mathbf{Z}}_j^{(\eta)}[\mathcal{I}_{j,obs}]$ for which $|\hat{z}_{i_m,j}^{(\eta)} - \hat{z}_{i,j}^{(\eta)}|$, $i_m \in \mathcal{I}_{j,mis}$, $i \in \mathcal{I}_{j,obs}$ is minimal. Suppose the d candidates are

$$\hat{z}_{i_1,j}^{(\eta)}, \hat{z}_{i_2,j}^{(\eta)}, \dots, \hat{z}_{i_d,j}^{(\eta)}, \quad i_1, i_2, \dots, i_d \in \mathcal{I}_{j,obs},$$

which can be written in the form of a set as $\mathcal{C}_{i,j}$, i.e., $\mathcal{C}_{i,j} = \{\hat{z}_{i_1,j}^{(\eta)}, \hat{z}_{i_2,j}^{(\eta)}, \dots, \hat{z}_{i_d,j}^{(\eta)}\}$. Finally, we obtain imputation for $z_{i,j}$, $i \in \mathcal{I}_{j,mis}$ by sampling from $\mathcal{C}_{i_m,j}$ and denote it as $z_{i_m,j}^{(\eta)}$, i.e.,

$$z_{i_m,j}^{(\eta)} \sim \mathcal{C}_{i_m,j}, \quad i_m \in \mathcal{I}_{j,mis}. \quad (20)$$

Indeed, the set $\mathcal{C}_{i_m,j}$ provides an empirical distribution for $z_{i_m,j}$, $i_m \in \mathcal{I}_{j,mis}$. The operations described from (17) to (20) correspond the conceptual description in (12) and (13). After all iterations, the final candidates corresponding to training dataset are denoted as $\hat{\mathbf{Z}}_j$, which are prepared for the use of sampling at the operational forecasting stage. In particular, missing values can be directly imputed via (18) when only point forecasts are needed.

3.3. Random forest

Indeed, the model described in (16) can be specified as any regression model, such as linear regression, random forest, etc. In this paper, it is specified as a random forest, as tree models usually perform well in practice (Januschowski et al., 2021). It grows B regression trees, each of which is trained on bootstrap samples from training data. Hence, the regression model that maps variables \mathbf{z}_{-j} to z_j is described as

$$g_j(\mathbf{z}_{-j}; \hat{\theta}_j) = \frac{1}{B} \sum_{b=1}^B g_{j,b}(\mathbf{z}_{-j}), \quad (21)$$

where $g_{j,b}(\mathbf{z}_{-j})$ is a regression tree. The splitting variable and splitting points of regression trees are often determined by the CART algorithm. Details about the CART algorithm can be found in (Hastie et al., 2001). Suppose we already have partitioned the variables into M regions, i.e., R_1, R_2, \dots, R_M . And we model the target as a constant c_m in each region. The regression function is described as

$$g_{j,b}(\mathbf{z}_{-j}) = \sum_{m=1}^M c_m I(\mathbf{z}_{-j} \in R_m), \quad (22)$$

where $I(\cdot)$ is the indicator function. In particular, c_m is estimated as the average of targets z_j in the region R_m , i.e.,

$$\hat{c}_m = \frac{1}{|\mathcal{I}_{R_m}|} \sum_{i \in \mathcal{I}_{R_m}} z_{i,j}, \quad (23)$$

where $\mathcal{I}_{R_m} = \{i \mid z_{i,-j} \in R_m\}$. The model grows like a binary tree. To begin with, we consider the space is split at variable Z_a , $a \in \{1, \dots, j-1, j+1, \dots, pk+1\}$ and point s , then we obtain two halves:

$$R_1(a, s) = \{z_{-j} \mid z_a \leq s\}, R_2(a, s) = \{z_{-j} \mid z_a > s\}. \quad (24)$$

It is fulfilled by a greedy algorithm, i.e.,

$$\min_{a,s} \left[\min_{c_1} \sum_{z_{i,-j} \in R_1(a,s)} \ell(z_{i,j} - c_1) + \min_{c_2} \sum_{z_{i,-j} \in R_2(a,s)} \ell(z_{i,j} - c_2) \right]. \quad (25)$$

Repeat the splitting process in the generated two regions, and stop only when minimum node size is reached.

3.4. Forecasting Stage

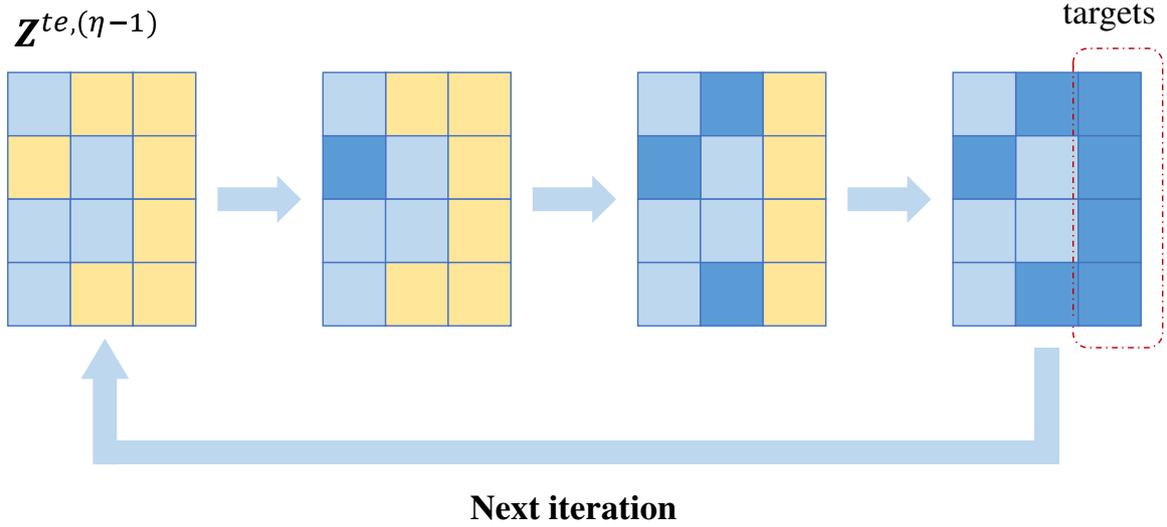


Figure 5: Illustration of the η -th iteration at the operational forecasting stage. Light blue blocks indicate observations, yellow blocks indicate missing values, and dark blue blocks indicate imputation (forecasting).

After training the imputation model, we obtain a collection of estimated random forests $\{g_j\}$ with parameters $\{\hat{\theta}_j\}$ and candidates $\{\hat{Z}_j\}$. At the operational forecasting stage, we feed sample $\mathbf{z}_t = [\mathbf{x}_t^\top, y_t]^\top$ (y_t is missing by default) into the estimated imputation model, and iteratively impute each missing value in \mathbf{z}_t according to (11), (13)-(15), which is illustrated in Figure 5. Compared to the training stage, parameters are fixed now; thus we only conduct iterative imputation here. Particularly, L equally likely imputations for \mathbf{z}_t are obtained, which are written as

$$\tilde{\mathbf{z}}_t^1, \tilde{\mathbf{z}}_t^2, \dots, \tilde{\mathbf{z}}_t^L.$$

Indeed, here $\mathbf{z}_t^{obs} = \mathbf{x}_t^{obs}$, $\mathbf{z}_t^{mis} = [\mathbf{x}_t^{mis\top}, y_t]^\top$. That is, \mathbf{z}_t^{mis} is imputed by realizations from the estimated distribution $f_{X^{mis}, Y|X^{obs}}(\mathbf{z}_t^{mis}, y_t | \mathbf{x}_t^{obs}; \mathcal{M}, \hat{\Theta})$. To get an empirical distribution for $f_{Y|X^{obs}}(y_t | \mathbf{x}_t^{obs}; \mathcal{M}, \hat{\Theta})$, we just fetch the corresponding value for y_t in each $\tilde{\mathbf{z}}_t^i$, i.e., the last entry of $\tilde{\mathbf{z}}_t^i$, which is denoted as \tilde{y}_t^i , i.e.,

$$\tilde{y}_t^i = \tilde{z}_{t, pk+1}^i, \quad i = 1, \dots, L. \quad (26)$$

Recall that y_t is the realization of the random variable $Y_{n,t+h}$, i.e., \tilde{y}_t^i is the realization from $f_{Y_{n,t+h}|t}(y | \mathbf{x}_t; \mathcal{M}, \hat{\Theta})$. Thus we rewrite \tilde{y}_t^i as $\tilde{y}_{n,t+h|t}^i$, all of which form a set, i.e.,

$$\{\tilde{y}_{n,t+h|t}^1, \tilde{y}_{n,t+h|t}^2, \dots, \tilde{y}_{n,t+h|t}^L\}.$$

Besides, we note that (26) is a surrogate of (10), which serves as marginalization operation when L is quite large. The point forecast $\hat{y}_{n,t+h|t}$ is given as an average, which is expressed as

$$\hat{y}_{n,t+h|t} = \frac{1}{L} \sum_{i=1}^L \tilde{y}_{n,t+h|t}^i. \quad (27)$$

4. Simulation study

Before validating the proposed approach on real data, we illustrate its applicability to point forecasting based on two related simulated processes, i.e., the autoregressive (AR) process and vector autoregressive (VAR) process. The results are assessed in terms of root-mean-square error (RMSE) here. Let $\mathcal{I}_{y,obs}$ denote the indices of observations in the test set. Then RMSE on test set is described as

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{I}_{y,obs}|} \sum_{t \in \mathcal{I}_{y,obs}} (y_t - \hat{y}_t)^2}, \quad (28)$$

where y_t denotes the observation at time t , \hat{y}_t denotes the point forecast at time t , and $|\mathcal{I}_{y,obs}|$ is the number of observed samples in the test set. In each case, we remove parts of generated data at random to simulate missingness, where the missing rate is varied from 5% to 50%. Situations where missing rates are larger than 50% are regarded impractical and thus not included in the study. Then, 80% of data are split as training set, whereas another 20% of data are split as test set for genuine forecasting validation. The missingness simulation and model validation are replicated 100 times for each missing rate.

4.1. AR process

In this case, we model an AR process of order 2, i.e.,

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \epsilon_t,$$

where α_0 is a constant, α_1 and α_2 are parameters, and ϵ_t is a white noise centered on 0. Let us set $\boldsymbol{\alpha}^\top$ as $[1, 0.33, 0.5]^\top$, and ϵ_t to follow the Gaussian $\mathcal{N}(0, 0.01)$. Concretely, we

simulate a time series of length 8760, corresponding to a year of data with 1-hour resolution, and present it in Figure 6 (a). The input features \mathbf{x}_t have 2 dimensions, and target y_t has 1 dimension. Specifically, the imputation model is trained by 10 iterations, as suggested by Van Buuren et al. (2006). At the operational forecasting stage, 200 realizations are generated in parallel, based on which point forecasts are derived according to (26) and (27). The RMSE values with respect to different missing rates are shown in Figure 6 (b). Intuitively, missing values lead to the increase of RMSE, and higher missing rates lead to larger RMSE.

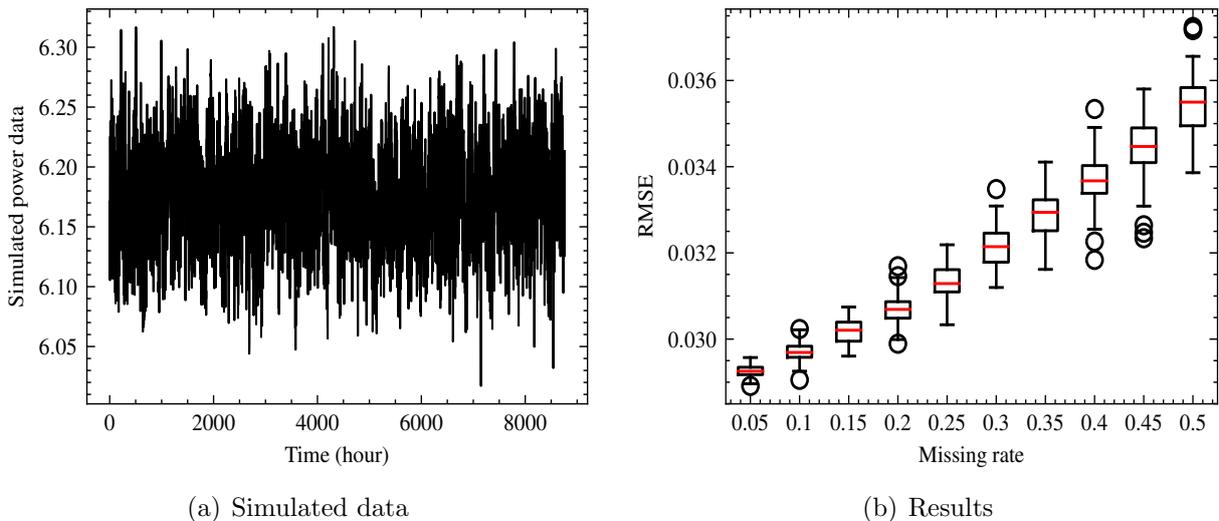


Figure 6: (a) Example simulated time-series with an AR process (single replicate); (b) Box plot of 1-step ahead RMSE in the presence of missing values based on AR simulated data with respect to different missing rates (based on Monte-Carlo simulations with 100 replicates).

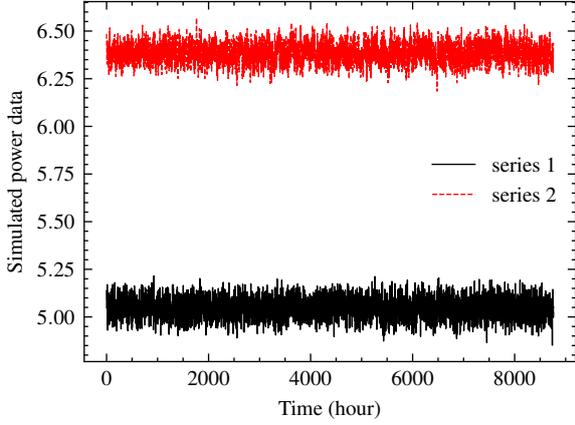
Since experiments at each missing rate are replicated 100 times, we obtain the variance of RMSE at each missing rate. As missing rate increases, the variance of RMSE also increases, because the influence on training varies to a larger extent when the missing rate is high.

4.2. VAR process

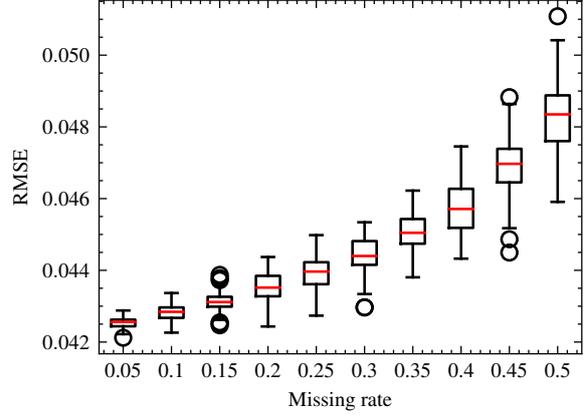
We model a VAR process of order 2, the simulation of which relies on 2 components, i.e., the auto-regressive structure of two time-series and their interdependence. We use two AR processes to generate intermediary series, and then perform a linear combination of the intermediary series. The intermediary series are simulated based on the following stochastic processes $Y_{A,t}$ and $Y_{B,t}$, i.e.

$$\begin{aligned} Y_{A,t} &= \alpha_{A,0} + \alpha_{A,1}Y_{A,t-1} + \alpha_{A,2}Y_{A,t-2} + \epsilon_{A,t}, \\ Y_{B,t} &= \alpha_{B,0} + \alpha_{B,1}Y_{B,t-1} + \alpha_{B,2}Y_{B,t-2} + \epsilon_{B,t}, \end{aligned}$$

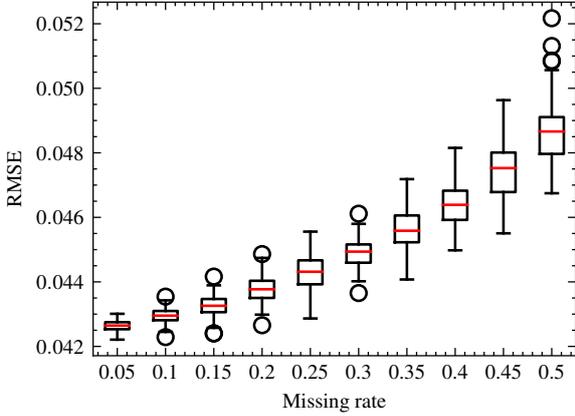
where $\boldsymbol{\alpha}_A^\top = [1, 0.88, -0.1]^\top$, $\boldsymbol{\alpha}_B^\top = [1, 0.9, -0.05]^\top$, $\epsilon_{A,t} \sim \mathcal{N}(0, 0.01)$, and $\epsilon_{B,t} \sim \mathcal{N}(0, 0.01)$. Then, the 2 final time series are obtained through a linear combinations of



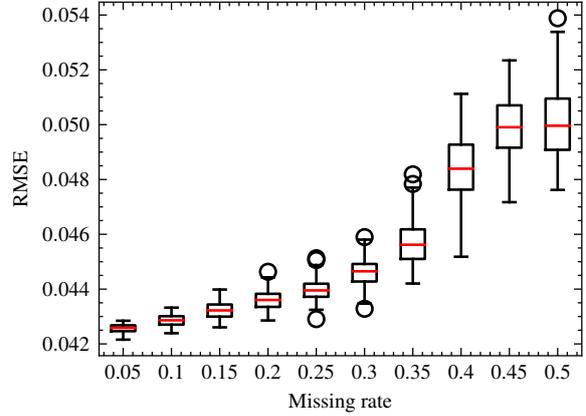
(a) Simulated data



(b) Features are from two series (no missing values in series 2)



(c) Features are only from series 1



(d) Features are from two series (with missing values in series 2)

Figure 7: (a) Simulated time series of the VAR process; (b) Box plot of 1-step ahead RMSE based on VAR simulated data with respect to different missing rates for series 1 (Monte-Carlo with 100 replications); (c) Box plot of 1-step ahead RMSE based on series 1 with respect to different missing rates for series 1 (Monte-Carlo with 100 replications); (d) Box plot of 1-step ahead RMSE based on VAR simulated data with respect to different missing rates for both series (Monte-Carlo with 100 replications).

$Y_{A,t}$ and $Y_{B,t}$, which are described as

$$\begin{aligned} Y_{1,t} &= \alpha_{1,1}Y_{A,t} + \alpha_{1,2}Y_{B,t}, \\ Y_{2,t} &= \alpha_{2,1}Y_{A,t} + \alpha_{2,2}Y_{B,t}, \end{aligned}$$

where $\alpha_1^\top = [0.9, 0.1]^\top$, $\alpha_2^\top = [0.3, 0.7]^\top$.

We still simulate time series of length 8760 and present them in Figure 7(a). In this case, we focus on forecasting future value of $Y_{1,t}$ by using previous realizations of both series. Now, the input features \mathbf{x}_t have elements, whereas the target y_t has a dimension of 1 only. We still train the imputation model with 10 iterations.

As a starting point, we assume there are no missing values in series $\{y_{2,t}\}$ and only vary the missing rate for $\{y_{1,t}\}$. The overall RMSE values are presented in Figure 7(b). As with the AR case, RMSE and its variance increases as the missing rate increases. For comparison, we consider two other scenarios, i.e., only using features from $\{y_{1,t}\}$, and simulating missingness for both $\{y_{1,t}\}$ and $\{y_{2,t}\}$, the results of which are respectively shown in Figure 7(c) and Figure 7(d). Comparing Figure 7(b) and Figure 7(c), we observe that the RMSE in Figure 7(b) is lower, which translates into saying that forecasting can be improved by using information from correlated series. Furthermore, as shown in Figure 7 (d), the benefit of using features from $\{y_{2,t}\}$ is still noticeable when the missing rate of $\{y_{2,t}\}$ is not too high. When the missing rate of $\{y_{2,t}\}$ is higher than 30%, using features of $\{y_{2,t}\}$ will even hamper the performance.

5. Case study

Besides the above simulation study, we further validate our approach based on real-world data from the USA. The case study considers a typical forecasting setup, where some data is used for estimating model parameters (training set) and the remainder of the data for genuine out-of-sample forecast verification (test set). Both point and probabilistic forecasting are considered. Also, since the dataset gathers data for multiple wind farms in a limited area, we can look at the case of employing univariate approaches (i.e., use of local data only), but also at a case where data from surrounding wind farms is used to improve forecasts. In that case, it is intuitively expected that one is further exposed to the likelihood and potential consequences of missing data. Note that the goal of this case study is not to pick and choose the best model and forecasting approach, but instead to show the impact of missing values on forecasting and the effectiveness of the proposed approach to accommodate those. In the following, we first describe the dataset and our experimental setup, the forecast verification framework and the benchmark approaches. The results obtained are then described and discussed.

5.1. Data description

Data from the USA are generated by the Wind Integration National Dataset (WIND) Toolkit (Draxl et al., 2015), which are therefore not completely real but capture the dynamics of wind power generation. Indeed, there are no missing values in this dataset. Then, we randomly remove some values to simulate missingness, based on which all models are estimated and validated. Concretely, the dataset contains 3 wind farms located in South Carolina, within a 150 km area. The spatial-temporal dynamics among wind farms suggests that one could use data of nearby wind farms to improve the forecasts. It gathers data over 7 years, from 2007 to 2013, with an hourly temporal resolution. All wind power measurements are normalized by their corresponding capacities.

5.2. Experimental setup

5.2.1. Different types of case-studies

Based on the data described in the above, we concentrate on both point and probabilistic forecasting in three different types of case-studies, representing alternative approaches to

forecasting (local data only, and with data from surrounding wind farms), as well as different types of missingness, i.e., sporadic and block missingness. We also consider forecasting with lead times from $k = 1$ to 6 steps ahead. More precisely these cases can be described as:

Case 1: Forecasting at a single site, using local data only (hence, with an autoregressive model). Data is missing sporadically and randomly, on a pointwise basis. The rates of missingness are of 10% and 20%, respectively, to investigate the performance of the approach conditional to how much data is missing.

Case 2: Forecasting at a single site using local data only (hence, with an autoregressive model). Data is missing over given time intervals (block missingness) though at random. The number of blocks with missing data is set to 600. These are randomly located over the dataset. The length of the block with missing data is random, and uniformly distributed between 5 to 30 time steps.

Case 3: Forecasting at a chosen site, but using data from both that sites and the nearby sites (hence, with a vector autoregressive model). The two types of missingness mentioned before (pointwise and block missingness) are considered.

In all 3 cases, when issuing a forecast at time t for lead time $t + k$, lagged observations are used as input features (since using autorregressive models). As feature selection is not the focus of this paper, we performed a preliminary study to select lags based on training data. As a result, we work in the following with autoregressive models with the 6 lagged observations (so, from $t - 5$ to t). The generalized logit-normal transform proposed by [Pinson \(2012\)](#) is further employed as a pre-processing stage to accommodate the double-bounded nature of wind power generation time-series (i.e., nonlinear and with the variance of residuals conditional upon the mean level).

5.2.2. Forecast verification: relevant scores and diagnostic tools

The quality of point forecasts is commonly evaluated with an RMSE criterion (consistent with the use of a quadratic loss in learning and forecast verification), whereas the quality of probabilistic forecasts is most often assessed by using the Continuous Ranked Probability Score (CRPS). Given a lead time h , we denote the cumulative density function for wind power generation Y_{t+h} , predicted at time t for time $t + h$, as F_{t+h} . Then, the CRPS for the predicted F_{t+h} and corresponding observation y_{t+h} is defined as

$$\text{CRPS}(F_{t+h}, y_{t+h}) = \int_y (F_{t+h}(y) - \mathbb{1}(y - y_{t+h}))^2 dy, \quad (29)$$

where $\mathbb{1}(\cdot)$ is a unit step function at the location of the observation y_{t+h} (also known as a Heaviside function), which can be regarded as the empirical cumulative density function of the observation y_{t+h} . Eventually, given the lead time h , we report the average CRPS value over all forecast-verification pairs, i.e.,

$$\text{CRPS}_h = \frac{1}{|\mathcal{I}_{y,obs}|} \sum_{t+h \in \mathcal{I}_{y,obs}} \text{CRPS}(F_{t+h}, y_{t+h}). \quad (30)$$

Besides the use of a proper skill score like the CRPS, informing about the overall skill and quality of the probabilistic forecasts (in the form of predictive densities), we will assess the probabilistic calibration of the predictive densities with reliability diagrams. For an extensive description of such reliability diagrams and their use in the assessment of probabilistic calibration, the reader is referred to [Pinson et al. \(2010\)](#). In parallel, in order to see how the probabilistic forecasts concentrate information, their sharpness is evaluated by calculated the width of central predictions intervals. I.e., for a given nominal coverage rate $1 - \beta$, these central prediction intervals are bounded by quantiles with nominal levels $\beta/2$ and $1 - \beta/2$. For a general overview of probabilistic forecast verification, see [Gneiting et al. \(2007\)](#).

5.2.3. Benchmarks

In general, we use three categories of benchmarks, i.e., climatology/persistence method, an ITP approach, and an UI approach with a distributional assumption. For point forecasting, persistence uses the latest observation as forecast. To implement the ITP approach, we respectively use mean imputation and advanced regression-based imputation namely MissForest ([Stekhoven and Bühlmann, 2012](#)) in the pre-processing procedure, and employ a random forest as the backbone regression model, which are abbreviated as RF-M and RF-R respectively. And, the state-of-the-art model DeepAR ([Salinas et al., 2020](#)) is adopted, which uses intermediate results of the long-short term memory model to impute missing values at both model estimation and operational stages. The copula-based imputation model proposed by [Zhao and Udell \(2020\)](#) is adopted to implement the UI approach. It is also a multiple imputation model, though relying on a distributional assumption. The retraining approach ([Tawn et al., 2020](#)) is used as a benchmark model, which consists in retraining the model without missing features. Besides, we consider a reference model that is implemented by a random forest and trained based on the complete dataset, which is abbreviated as RF-C. The benchmark models for point forecasting, as well as used abbreviations, are gathered in Table 1.

Table 1: Abbreviations for point and probabilistic forecasting benchmark models.

Abbreviation	Description (point forecasting)
RF-M	Random forest with the mean imputation as preprocessing
RF-R	Random forest with the regression-based imputation as preprocessing
Copula	Copula-based imputation model within universal imputation strategy
DeepAR	Deep learning model that uses intermediate results to impute missing values
RF-C	Random forest trained based on the complete dataset
Abbreviation	Description (probabilistic forecasting)
Gauss-M	Gaussian model with the mean imputation as preprocessing
Gauss-R	Gaussian model with the regression-based imputation as preprocessing
Copula	Copula-based imputation model within universal imputation strategy
DeepAR	Deep learning model that uses intermediate results to impute missing values
QR-R	QR model with the regression-based imputation as preprocessing
QR-C	QR model trained based on the complete dataset

As for probabilistic forecasting, the climatology is set as a naive benchmark. It utilizes

the empirical distribution of all historical values to communicate the probability distribution of future wind power generation. To implement the ITP approach, a model with the Gaussian distributional assumption as well as a QR model are adopted as backbone models. In particular, the base model chosen for QR is the gradient boosting machine (Landry et al., 2016), which won the GEFCom 2014 for instance. For the model with the Gaussian distributional assumption, we use a neural network to estimate the shape parameters of Gaussian distributions. The QR model with regression-based imputation as preprocessing is abbreviated as QR-R, while the Gaussian models with mean and regression-based imputation are abbreviated as Gauss-M and Gauss-R. Again, the ‘DeepAR’ model (Salinas et al., 2020) is used as a benchmark, since it is allowed to communicate Gaussian densities. The UI approach is still implemented via the copula-based model. Besides, we set the QR model trained based on the complete dataset as a reference, which is abbreviated as QR-C. The benchmark models for probabilistic forecasting, as well as corresponding abbreviations, are also collated in Table 1.

5.3. Results and discussion

Results that correspond to the aforementioned three cases are respectively reported in three different subsections and followed by further discussion.

5.3.1. Case 1

Emphasis is first placed on sporadic missingness, i.e., for the case where single values are missing, at random times. Let us start by presenting and discussing results for the most severe rate of missingness, of 20%. In practice, this means that 20% of the values are missing, at random locations over both training and testing sets. The point forecasting results in terms of RMSE are collated in Table 2.

Table 2: RMSE values as a function of the lead time (Case 1, missing rate of 20%). RMSE values are expressed in percentage of normalized capacity.

Lead Time	Persistence	RF-M	RF-R	Copula	FCS	DeepAR	RF-C
1	16.8	17.7	16.1	17.3	15.9	17.0	14.6
2	21.1	20.9	19.8	21.3	19.5	20.6	18.9
3	24.7	23.4	22.5	24.3	22.3	23.3	21.9
6	32.7	28.2	28.0	30.6	27.9	29.3	27.6

Not surprisingly, the RMSE increases with the lead time, and, forecast quality in the presence of missing values is worse than when there is no missing data. There, missing values have a negative impact at both model estimation and operational forecasting stages. Persistence is a competitive benchmark, as it is easy to implement and the resulting forecast quality is difficult to outperform for such short lead times. In parallel, RF-M performs worse than persistence for 1-step ahead forecasts, most likely due to errors in imputation introduced by this pre-processing procedure. Given training datasets \mathbf{X}^{tr} and \mathbf{Y}^{tr} , one can estimate a regression model f^P that is equivalent to the reference model, if the imputed datasets are as same as the real complete datasets $\mathbf{X}^{tr,P}$ and $\mathbf{Y}^{tr,P}$. However, the imputed

datasets are usually deviated from the real complete datasets. Then, the model estimated based on $\mathbf{X}^{tr,C}$ and $\mathbf{Y}^{tr,C}$, denoted as f^C , is different from f^P . That is, the closer the imputed datasets are to the real complete datasets, the closer f^C is to f^P . Obviously, RF-R has better performance than RF-M, since the regression-based imputation is superior to the mean imputation. Besides, at the operational forecasting stage, it is still required to impute input features, which may also accumulate errors. Although DeepAR is free of any pre-processing stage, the imputed values may still deviate from real values, introducing errors to both the model training and forecasting stages. As shown in Table 2, the performance of DeepAR is even worse than the simple benchmark model, i.e., RF-R.

The used copula- and FCS- based models fall into the category of UI approach. Compared to the ITP approach, the UI approach has an advantage that it is free of a pre-processing procedure, which avoids introducing errors aroused by the pre-processing procedure into the forecasting task. The FCS-based model outperforms RF-M and RF-R, while the copula-based model is inferior to them, as revealed in Table 2. Although copula method allows to characterize several kinds of distributions, it is required to specify the transform function here, which means that a specific distributional assumption is implied. This may impede the performance of copula-based model when the distributional assumption cannot fit the underlying distribution well. By contrast, the FCS method is free of such an assumption, and therefore has a better performance than the copula-based model. It suggests that by using distribution-free imputation method like FCS, the UI approach is superior to ITP approach.

Besides, we compare our proposed approach and the retraining approach discussed by Tawn et al. (2020) by focusing on a specific missing pattern (i.e., the last feature is missing), and present the RMSE values in Table 3. It is intuitive that the performance of the retraining approach is comparable to RF-M and FCS, as the retraining approach is free of a pre-processing stage and utilizes a complete set of observations to estimate the parameters. Eventually, the quality of the forecasts is linked to the informative value of the features retained. However, it also implies that a specific model is needed for each missingness pattern. Then, a specific training dataset is required for each pattern, which means only parts of the data are used to estimate a model. Besides, the retraining approach will suffer the curse of dimensionality. That is, denoting the dimension of features as d , the retraining approach will independently train 2^d models. While the training time for a set of point forecasting models may be acceptable, the computational costs will steeply increase for probabilistic forecasting case. In contrast, the proposed UI approach is not only free of any pre-processing stage, but also applicable to all missingness patterns once trained.

Table 3: RMSE for 1-step ahead forecasts (Case 1, last feature missing). RMSE values are expressed in percentage of normalized capacity.

Lead Time	Persistence	RF-M	RF-R	Copula	FCS	DeepAR	Retraining
1	15.8	16.2	15.1	15.9	15.0	15.8	15.3

Next, we move on to the results for probabilistic forecasting, with the CRPS values obtained collated in Table 4. Here, missing values have nearly no influence on the performance

of climatology, since climatology characterizes uncertainty based the empirical distribution of all historical observations. This distribution is not highly modified when a fairly limited number of samples are missing.

Table 4: CRPS as a function of the lead time (Case 1, missing rate of 20%). CRPS values are expressed in percentage of normalized capacity.

Lead Time	Climatology	Gauss-M	Gauss-R	QR-R	Copula	FCS	DeepAR	QR-C
1	18.6	9.2	7.5	7.8	11.5	6.9	7.8	6.9
2	18.6	11.2	9.9	9.9	14.6	9.1	10.2	9.3
3	18.6	12.7	11.7	11.7	17.0	10.9	12.1	11.2
6	18.6	15.9	15.5	15.4	22.4	14.7	16.5	15.1

Comparing the Gauss-M and Gauss-R, we know that a better imputation method is still preferred by the ITP strategy in the context of probabilistic forecasting. Both the Gauss-R and QR-R use the regression-based imputation as preprocessing procedure. But they differ in backbone models – Gauss-R relies on the Gaussian distributional assumption, whereas QR-R is distribution-free. Their performance is comparable in this case, which is different from the usual situation (i.e., complete datasets) where QR is always superior. Obviously, one needs to estimate shape parameters of Gaussian distribution in Gauss-R, but the parameters of several quantile functions in QR-R. The parallel estimation of QR-R models may result in more errors in the ultimate estimated distribution. Therefore, results are governed by both models and the influence of missing values on model estimation. The performance of DeepAR is slightly worse than that of Gauss-R and QR-R, which suggests handling missing values in forecasting is nontrivial. Values imputed by the intermediate results of the model may also introduce errors at the model estimation stage. The FCS-based model outperforms Gauss-R and QR-R, whereas the performance of copula-based model is worse than those of Gauss-R and QR-R, which suggests that the distributional assumption may impede the performance of UI approach. Besides, the performance of FCS-based model is comparable to that of the reference QR model trained based on complete dataset, which validates the effectiveness of FCS-based model.

We present the 90% PIs of 6 days issued by the FCS-based and reference models, respectively, in Figure 8.

Although the FCS-based approach encounters missing values at both model estimation and operational forecasting stages, its PIs are similar to those of the reference model. Specifically, at some periods e.g. from 35-h to 45-h, the prediction interval issued by the FCS-based approach are actually sharper than those of the reference model. A reliability assessment through the use of reliability diagrams is given in Figure 9(a), while a sharpness assessment is performed by looking at the width of central prediction intervals (as a function of their nominal coverage rate), and depicted in Figure 9(b). Models based on an ITP strategy tend to underestimate lower quantiles, while the reliability of DeepAR and copula based model deviates to the ideal case to some extent. The FCS-based model achieves level of reliability and sharpness that are comparable to the reference model. The average of absolute values of deviations from perfect reliability is shown in Table 5. The deviation of the FCS is

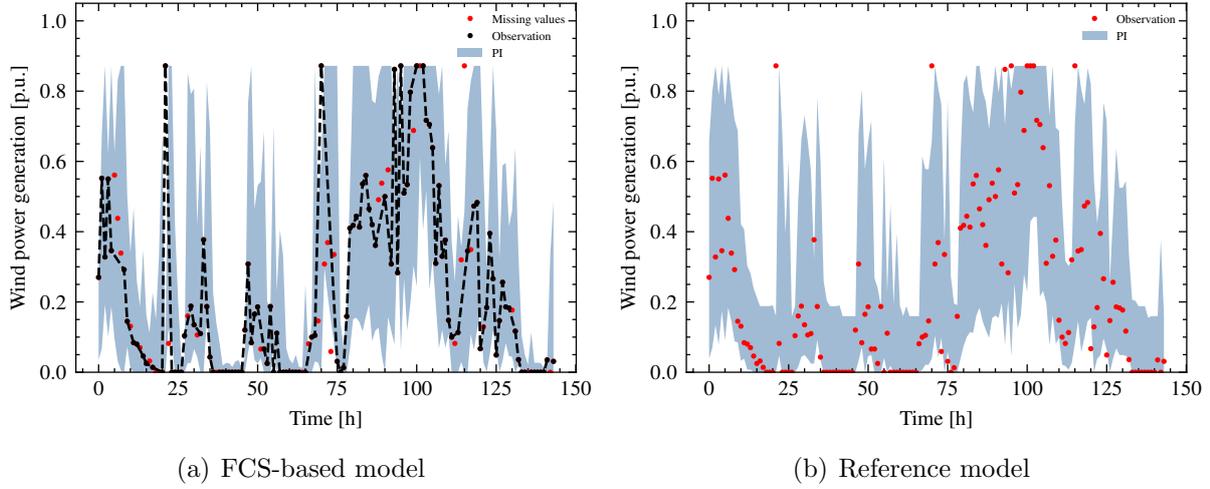


Figure 8: Illustration of 1-step ahead 90% central prediction intervals over a period of 6 days, as issued by the FCS-based model (a), and the reference model (b).

even smaller than that of reference model, which is likely due to that the FCS is robust to overfitting.

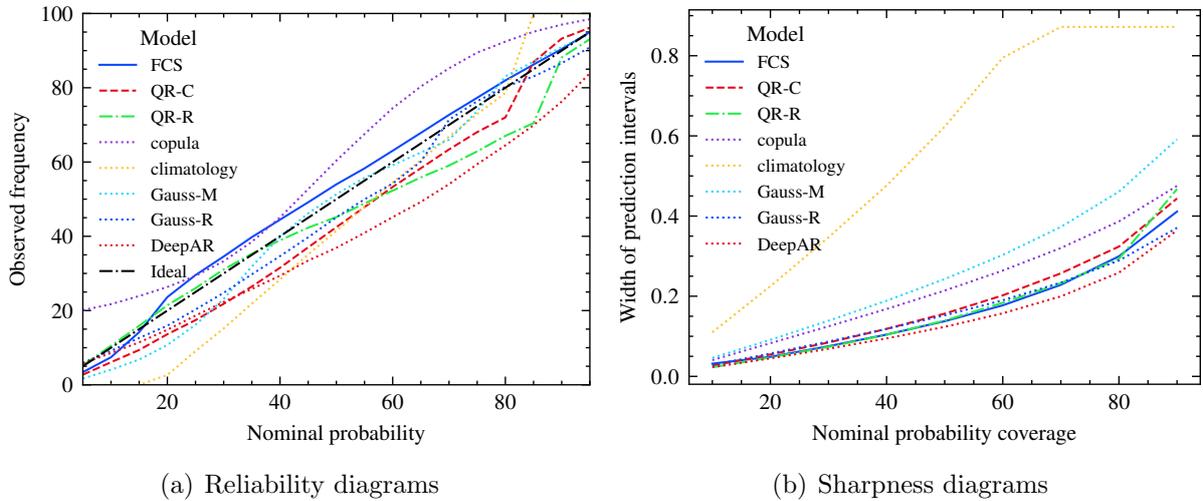


Figure 9: Assessment of 1-step ahead probabilistic forecasts for all models for Case 1, based on reliability diagrams (a) and sharpness diagrams (b).

Table 5: The average of absolute values of deviations from perfect reliability in Case 1 (in percent).

Lead Time	Climatology	Gauss-M	Gauss-R	QR-R	Copula	FCS	DeepAR	QR-C
1	9.20	3.37	3.52	4.81	4.81	2.78	10.57	6.04

The RMSE and CRPS values, when considering a missing rate of 10%, are collated in

Table 6 and Table 7, respectively. Compared to the results with a missing rate of 20%, the quality of the forecasts is improved. For point forecasting, the performance of RF-R is comparable to that of FCS, which means that the ITP strategy may be more acceptable when the missing rate is not that high. This is while, in the context of probabilistic forecasting, FCS still outperforms other models, which suggests that missing values may pose greater challenges to probabilistic forecasting. Besides, the performance of FCS is even better than that of QR-C, possibly hinting at the fact that FCS is less prone to overfitting.

Table 6: RMSE values with different lead times in Case 1 on the condition that missing rate is 10% (percentage of normalized capacity).

Lead Time	Persistence	RF-M	RF-R	Copula	FCS	DeepAR	RF-C
1	15.9	15.9	15.2	16.6	15.1	16.2	14.6
2	20.6	19.7	19.2	20.8	19.1	20.2	18.9
3	24.4	22.5	22.1	24.1	22.0	23.3	21.9
6	32.5	27.9	27.8	30.5	27.8	29.5	27.6

Table 7: CRPS values with different lead times in Case 1 on the condition that missing rate is 10% (percentage of normalized capacity).

Lead Time	Climatology	Gauss-M	Gauss-R	QR-R	Copula	FCS	DeepAR	QR-C
1	18.6	8.1	7.2	7.4	11.2	6.6	7.4	6.9
2	18.6	10.5	9.6	9.6	14.3	8.9	9.9	9.3
3	18.6	12.1	11.5	11.5	16.9	11.9	9.7	11.2
6	18.6	15.9	15.3	15.3	22.4	14.7	16.6	15.1

5.3.2. Case 2

In contrast to the sporadic missingness of Case 1, we simulate here missing values that span over time intervals (hence, referred to as block missingness). Remember that 600 blocks are randomly spread over the whole dataset, with length between 5 and 30 time steps. Let us first analyse and discuss point forecasting results. As a basis, the RMSE values of the points forecasts obtained with the different approaches are gathered in Table 8.

Table 8: RMSE values with different lead times in Case 2 (percentage of normalized capacity).

Lead Time	Persistence	RF-M	RF-R	Copula	FCS	DeepAR	RF-C
1	15.8	14.9	14.8	16.4	14.9	15.9	14.6
2	20.9	19.2	19.2	21.1	19.3	21.0	18.9
3	24.8	22.2	22.2	24.3	22.4	24.1	21.9
6	32.9	27.9	27.9	30.6	28.1	30.9	27.6

The performance of RF-M is comparable to that of RF-R, most likely due to the fact that most samples here are complete. In contrast, the copula-based model performs much worse than both RF-M and RF-R. Indeed, the estimation stage for the copula-based model is

based on an expectation-maximization algorithm, which is sensitive to samples whose values are entirely missing. This may suggest that the copula-based model is not applicable to the situation of block missingness. Certainly, the samples whose values are entirely missing contain no information, and can be deleted at the model estimation stage. As with Case 1, the performance of DeepAR is worse than that of RF-R, which reveals a caveat for the existing DeepAR framework in handling missing values. The performance of the FCS-based approach is comparable to that of RF-M/RF-R. One may then infer that ITP and UI types of strategies perform fairly similar for point forecasting when experiencing block missingness. However, the picture will look different when extending the study to probabilistic forecasting. To assess the performance of the various approaches for that probabilistic forecasting case, we first look at CRPS values, which are gathered in Table 9.

Table 9: CRPS values with different lead times in Case 2 (percentage of normalized capacity).

Lead Time	Climatology	Gauss-M	Gauss-R	QR-R	Copula	FCS	DeepAR	QR-C
1	18.6	7.0	6.9	7.2	11.2	6.5	7.1	6.9
2	18.6	9.8	9.6	9.7	14.5	9.0	10.2	9.3
3	18.6	11.8	11.8	11.6	17.1	10.9	12.6	11.2
6	18.6	15.7	15.7	15.6	22.4	14.9	17.9	15.1

Not surprisingly, the performance of Gauss-R is slightly superior to that of Gauss-M. But their difference is smaller than what was observed in Case 1. It could be inferred that it is difficult to handle block missingness via imputation techniques. In the context of block missingness, regression-based imputation will also tend to impute missing values with mean values. Still, the performance of Gauss-R is comparable to that of QR-R. The FCS-based model yields the best performance. Combined to the results of Case 1, it indicates that this approach seems to be superior for different types of missingness, here both sporadic and block missingness. A clear point is that ITP strategies are highly sensitive to samples whose values are entirely missing, since the rationale of ITP strategies is to utilize observed parts of samples to infer the missing parts. If a sample is completely unobserved, no information could be used for learning and eventually forecasting.

Both reliability and sharpness are evaluated in Figure 10, for 1-step ahead probabilistic forecasts (with reliability diagrams in Figure 10(a) and sharpness diagrams in Figure 10(b)). The FCS-based approach achieves acceptable probabilistic calibration, especially for the case of lower and higher quantiles. As summary statistics, the average deviation (in absolute value) for perfect reliability is given in Table 10, for all approaches. There again, one verifies that the FCS-based approach yields the lowest deviation. In parallel, the prediction interval width for QR, Gaussian-based and FCS-based approaches are very close, for all nominal coverage rates. The prediction interval width for DeepAR is somewhat smaller, though at the price of poorer probabilistic calibration. This is also reflected by the larger CRPS values for DeepAR, compared to the FCS-based approach.

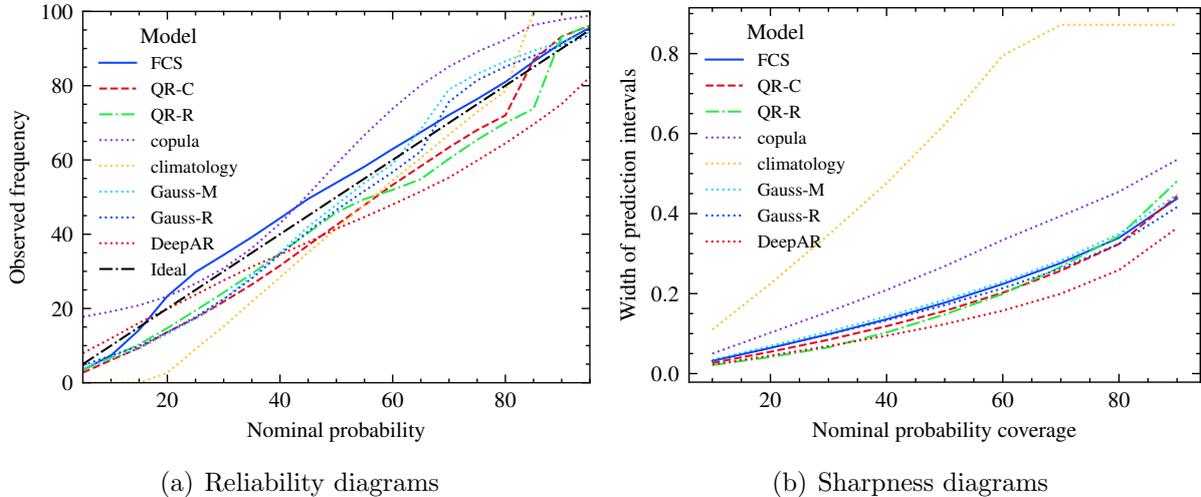


Figure 10: Assessment of 1-step ahead probabilistic forecasts for all models for Case 2, based on reliability diagrams (a) and sharpness diagrams (b).

Table 10: Average deviation (in absolute value) from perfect reliability in Case 2 (in percent).

Lead Time	Climatology	Gauss-M	Gauss-R	QR-R	Copula	FCS	DeepAR	QR-C
1	9.21	4.47	4.27	6.01	8.29	2.69	8.37	6.04

5.3.3. Case 3

In this subsection we show that forecasting in the presence of missing values can still be improved by utilizing information of nearby sites as auxiliary features (AFs). Besides input features of the chosen wind farm, we use previous wind power generation values from two nearby wind farms as AFs. It is assumed that the missingness of nearby wind farms is different from the target wind farm, which is practical since missingness is usually caused by sensor faults or communication errors. We consider both the sporadic missingness and block missingness here. Particularly, we concentrate on 1-step ahead forecasts and investigate the impacts of different missing rates or missing blocks in AFs. The RMSE values in the context of sporadic missingness are shown in Figure 11(a) ('AFs $p\%$ m' means $p\%$ of auxiliary features are missing), where we simulate different missing rates at two nearby wind farms and set the missing rate at the target wind farm as 20%.

As expected, the accuracy of point forecasting is improved with the assistance of AFs, which is comparable to RF-C in the Table 1. Furthermore, it can be seen that the benefit of AFs is robust, since the performance is relatively consistent as the missing rate of AFs increases. It might be explained by the fact that the key information for forecasting comes from the target wind farm itself. So, it may not make a big difference when few auxiliary features are missing. The results of probabilistic forecasting is also shown in Figure 11(a), which also suggests that AFs provide extra information and thus contribute to improving probabilistic forecasts. The RMSE and CRPS values in the context of block missingness are presented in Figure 11(b) ('AFs c m' means there are c missing blocks in auxiliary features),

where we simulate 600 missing blocks at the target wind farm. It is seen that auxiliary features can still improve the quality of the forecasts in that case.

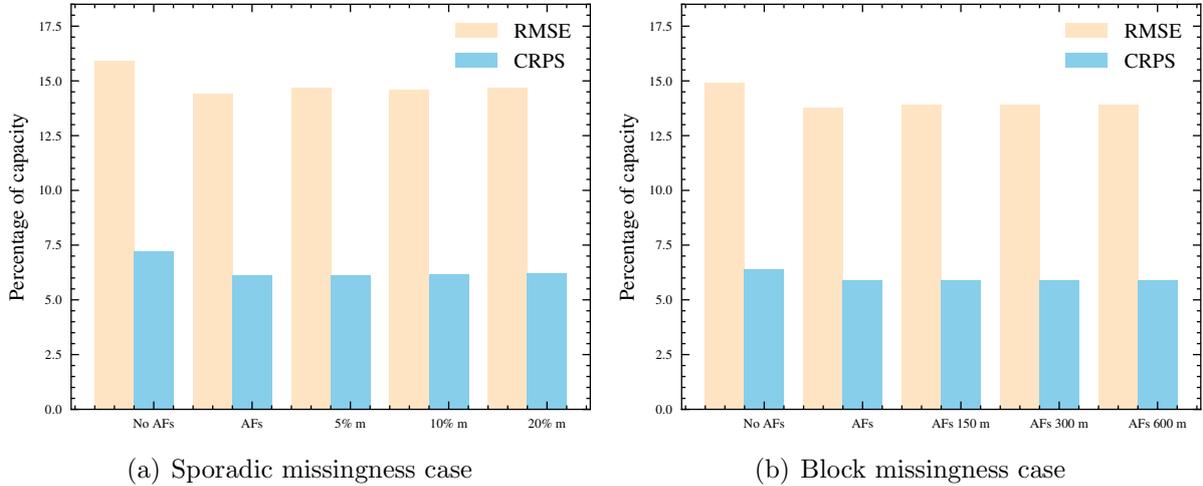


Figure 11: Assessment of 1-step ahead probabilistic forecasts for Case 3, in context of sporadic missingness (a) and block missingness (b).

5.3.4. Training time

We note that by using the FCS method, the proposed UI approach is always superior in the context of probabilistic forecasting. However, it costs much time to perform Gibbs sampling to provide probabilistic forecasting. The computation will significantly increase when the dimension of variable gets larger. We present the training time and operational time in Table 11 for illustration. As shown, the training time of FCS is larger than that of QR models, but manageable compared to that of DeepAR. Therefore, it is required to find computationally efficient methods to implement the proposed approach. As the fully conditional specification method iteratively estimates several conditional distribution models, it is hard to further reduce the training time. But it is feasible to directly learn the joint probability distribution model via a joint modeling approach, which would considerably reduce the training time.

Table 11: Training time and operational time for 1-step probabilistic forecasting in Case 1.

	Gauss-R	QR-R	Copula	FCS	DeepAR
Training time (min)	32	1	9	41	67
Operational time (s)	$\ll 0.01$	$\ll 0.01$	$\ll 0.01$	0.01	0.01

6. Conclusions

It is intuitive to want to consider an “impute, then predict” approach to deal with missing values, as existing forecasting methods can be readily used after the (imputing)

pre-processing procedure. However, while such a pre-processing procedure at the model estimation stage jointly imputes input features and targets, it only imputes input features at the operational forecasting stage, possibly in a way that is not consistent with the model used for forecasting eventually. In this paper instead, we propose a “universal imputation” approach, motivated by the problem of wind power forecasting in the presence of missing values. As for many other application areas, it is very common to have missing values within wind power forecasting. Our proposal approach relies on multiple imputation methods, and jointly performs the imputation of missing values of input features and the forecasting of targets. That is, it does not require a pre-processing procedure, while being consistent through model estimation and operational forecasting stages. Under the assumption such that observations are missing at random, parameters can be estimated based on observations only, at the model estimation stage. At the operational stage, it treats targets as missing values, and iteratively impute both the missing values of input features and targets. Particularly, as multiple imputation provides several realizations from the joint distribution of input features and targets, the proposed approach naturally allows to issue both point and probabilistic forecasts. The case studies based on WIND Toolkit (over the USA) confirm the applicability of this approach. Not surprisingly, forecast quality necessarily decreases as the missing rate of dataset increases. The results also suggest that the FCS-based method performs better than the “impute, then predict” approach; it is especially preferred in the probabilistic forecasting case. And, the results suggest that the FCS-based approach may prevent overfitting in some extent. It also further validates the benefits from sharing information and data among wind farms, even in the presence of missing values.

We note that the modeling approach is quite different from the commonly used forecasting approaches in the context of complete datasets. The goal of this paper is not to replace the existing approaches, but to offer a complementary tool for use in the presence of missing values. We also expect there are similar ways to generalize commonly used modelling and forecasting approaches to the case of missing data. The computational costs of the introduced FCS-based approach are high, and growing significantly as the dimension increases. Therefore, more efficient methods are still needed. It may be appealing to alternate the FCS method with distribution-free joint modeling imputation approaches. Our proposal is based on the “missing-at-random” assumption, and thus avoids modeling the distribution of missingness. The situation where observations are missing not at random should be further explored in the future. Besides, emphasis should be placed on relaxing the stationary assumption in order to deal with non-stationary environments, e.g., with online learning.

References

- Benidis, K., Rangapuram, S.S., Flunkert, V., Wang, Y., Maddix, D., Turkmen, C., Gasthaus, J., Bohlke-Schneider, M., Salinas, D., Stella, L., Aubet, F.X., Callot, L., Januschowski, T., 2022. Deep learning for time series forecasting: Tutorial and literature survey. *ACM Computing Surveys* .
- Cao, W., Wang, D., Li, J., Zhou, H., Li, Y., Li, L., 2018. Brits: bidirectional recurrent imputation for time series, in: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 6776–6786.
- Cavalcante, L., Bessa, R.J., Reis, M., Browell, J., 2017. Lasso vector autoregression structures for very short-term wind power forecasting. *Wind Energy* 20, 657–675.

- Che, Z., Purushotham, S., Cho, K., Sontag, D., Liu, Y., 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports* 8, 1–12.
- De Gooijer, J.G., et al., 2017. *Elements of nonlinear time series analysis and forecasting*. volume 37. Springer.
- Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39, 1–22.
- Draxl, C., Clifton, A., Hodge, B.M., McCaa, J., 2015. The wind integration national dataset (wind) toolkit. *Applied Energy* 151, 355–366.
- Gneiting, T., Balabdaoui, F., Raftery, A.E., 2007. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society, Series B* 69, 243–268.
- Golyandina, N., Osipov, E., 2007. The “caterpillar”-ssa method for analysis of time series with missing values. *Journal of Statistical planning and Inference* 137, 2642–2653.
- Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. MIT Press.
- Hastie, T., Tibshirani, R., Friedman, J., 2001. *The Elements of Statistical Learning*. Springer Series in Statistics, Springer New York Inc., New York, NY, USA.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural computation* 9, 1735–1780.
- Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., Hyndman, R.J., 2016. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. *International Journal of Forecasting* 32, 896–913.
- Hong, T., Pinson, P., Wang, Y., Weron, R., Yang, D., Zareipour, H., 2020. Energy forecasting: A review and outlook. *IEEE Open Access Journal of Power and Energy* .
- Hyndman, R.J., Khandakar, Y., 2008. Automatic time series forecasting: the forecast package for r. *Journal of statistical software* 27, 1–22.
- Januschowski, T., Wang, Y., Torkkola, K., Erkkilä, T., Hasson, H., Gasthaus, J., 2021. Forecasting with trees. *International Journal of Forecasting* .
- Jones, R.H., 1980. Maximum likelihood fitting of arma models to time series with missing observations. *Technometrics* 22, 389–395.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y., 2017. Lightgbm: a highly efficient gradient boosting decision tree, in: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3149–3157.
- Koenker, R., Hallock, K.F., 2001. Quantile regression. *Journal of economic perspectives* 15, 143–156.
- Kohn, R., Ansley, C.F., 1986. Estimation, prediction, and interpolation for arima models with missing data. *Journal of the American statistical Association* 81, 751–761.
- Landry, M., Erlinger, T.P., Patschke, D., Varrichio, C., 2016. Probabilistic gradient boosting machines for gefcom2014 wind forecasting. *International Journal of Forecasting* 32, 1061 – 1066.
- Little, R.J., Rubin, D.B., 2019. *Statistical analysis with missing data*. volume 793. John Wiley & Sons.
- Liu, T., Wei, H., Zhang, K., 2018. Wind power prediction with missing data using gaussian process regression and multiple imputation. *Applied Soft Computing* 71, 905–916.
- Messner, J.W., Pinson, P., 2019. Online adaptive lasso estimation in vector autoregressive models for high dimensional wind power forecasting. *International Journal of Forecasting* 35, 1485–1498.
- Montero-Manso, P., Hyndman, R.J., 2021. Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting* 37, 1632–1653.
- Pinson, P., 2012. Very-short-term probabilistic forecasting of wind power with generalized logit–normal distributions. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 61, 555–576.
- Pinson, P., McSharry, P., Madsen, H., 2010. Reliability diagrams for non-parametric density forecasts of continuous variables: Accounting for serial correlation. *Quarterly Journal of the Royal Meteorological Society* 136, 77–90.
- Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T., 2020. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 1181–1191.
- Sangnier, M., Fercoq, O., d’Alché Buc, F., 2016. Joint quantile regression in vector-valued rkhs, in: *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 3700–3708.

- Stekhoven, D.J., Bühlmann, P., 2012. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics* 28, 112–118.
- Stone, C.J., 1991. Asymptotics for doubly flexible logspline response models. *The annals of statistics* 19, 1832–1854.
- Tawn, R., Browell, J., Dinwoodie, I., 2020. Missing data in wind farm time series: Properties and effect on forecasts. *Electric Power Systems Research* 189, 106640.
- Van Buuren, S., 2018. *Flexible imputation of missing data*. CRC press.
- Van Buuren, S., Brand, J.P., Groothuis-Oudshoorn, C.G., Rubin, D.B., 2006. Fully conditional specification in multivariate imputation. *Journal of statistical computation and simulation* 76, 1049–1064.
- Wan, C., Lin, J., Wang, J., Song, Y., Dong, Z.Y., 2016. Direct quantile regression for nonparametric probabilistic forecasting of wind power generation. *IEEE Transactions on Power Systems* 32, 2767–2778.
- Wen, H., Pinson, P., Ma, J., Gu, J., Jin, Z., 2022. Continuous and distribution-free probabilistic wind power forecasting: A conditional normalizing flow approach. *arXiv preprint arXiv:2206.02433* .
- You, J., Ma, X., Ding, Y., Kochenderfer, M.J., Leskovec, J., 2020. Handling missing data with graph representation learning, in: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (Eds.), *Advances in Neural Information Processing Systems*, pp. 19075–19087.
- Zhao, Y., Udell, M., 2020. Missing value imputation for mixed data via gaussian copula, in: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 636–646.