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 like fully conditional specification, the proposed approach is superior to the common 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 investigation 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., 2021) 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, a natural 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.

In addition to the aforementioned ITP approach, 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 located at the end of a sequence. Furthermore, both imputation and forecasting tasks assume the continuation of the underlying structure of data, and 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 be then left aside when inferring the underlying structure of interest. Consequently, the problem boils down to estimating the parameters of a model based on incomplete observations only. With this idea, 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 this paper, we model this problem in the probabilistic setting and focus on the application to very-short term WPF where missing value issues often occur.

Input features and targets follow an underlying joint multivariate distribution. 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. Concretely, by setting

the imputation method as multiple imputation (Dempster et al., 1977), missing values can be imputed with several equally likely realizations from the distribution, which therefore 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 both the USA and Denmark. The results show that by choosing imputation methods free of distributional assumptions, the proposed approach is superior to existing ITP approaches. The main contributions of this paper are two-fold. One of them is a proposal for wind power forecasting in the presence of missing values, which jointly accommodate imputation and forecasting tasks within the universal multiple imputation framework. The other contribution is that we have shown its applicability to wind power forecasting in both point and probabilistic setting.

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 $(Y_1, \cdots, Y_{10})$. 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 $\boldsymbol{Z}$ represent a matrix, $\mathcal{I}$ and $\mathcal{J}$ denote row indices and column indices. Then, $\boldsymbol{Z}[\mathcal{I}; \mathcal{J}]$ represents a part of matrix $\boldsymbol{Z}$ indexed by $\mathcal{I}$ and $\mathcal{J}$. And, $(\cdot)^\top$ denotes the transpose of matrices. A time series is represented as $\{y_t, t = 1, 2, \cdots\}$ indexed by time t, which is a realization of a stochastic process $\{Y_t, t = 1, 2, \cdots\}$. We also write them as $\{y_t\}$ and $\{Y_t\}$ for short.

2. Preliminaries

In this section, first we formulate the very-short term wind power forecasting problem, and discuss the missing value issue. Then we introduce the challenges brought 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 represents the 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., $\Omega_t = (\Omega_{1,t}, \cdots, \Omega_{p,t})$. Generally, the aim is to issue forecasts with lead time h, i.e., the characteristics of $Y_{1,t+1}, \cdots, Y_{1,t+h}, \cdots, Y_{p,t+1}, \cdots, Y_{p,t+h}$, 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

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

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 $\{Y_{1,t}, Y_{2,t}, \cdots, Y_{p,t}\}$ is stationary. Then, the density function $f_{Y_{1,t},\cdots,Y_{p,t+h}}$ 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

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

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

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

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

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

In very-short term WPF, one usually uses wind power generation values of length k up to time t as input features, i.e., $\Omega_{n,t} = (y_{n,t-k+1}, \cdots, y_{n,t})$, which is also written as a vector $[y_{n,t-k+1}, \cdots, y_{n,t}]^\top \in [0, P_n]^k$ for computation. Therefore, Ω_t is represented as the vector

$$[y_{1,t-k+1}, \cdots, y_{1,t}, \cdots, y_{p,t-k+1}, \cdots, y_{p,t}]^\top \in [0, P_1]^k \times [0, P_2]^k \times \cdots \times [0, P_p]^k.$$

For simplicity of notations, let us focus on forecasting $\hat{f}_{n,t+h|t}$ given the features Ω_t at time t, and respectively denote the features and the realization of target $Y_{n,t+h}$ as $\boldsymbol{x}_t$ and y_t. Assume the time series $\{y_{1,t}, y_{2,t}, \cdots, y_{p,t}\}$ up to time T is available, and thus one can get N sample pairs $(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_N, y_N)$. They can be written in the form of a matrix, i.e., $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]^\top$ as well as $\boldsymbol{Y} = [y_1, \cdots, y_N]^\top$. The matrix $\boldsymbol{X}$ is of shape $N \times pk$, whereas $\boldsymbol{Y}$ is of shape $N \times 1$. Now the density forecast described in (3) boils down to conditional probability density function estimation.

Based on a stationarity assumption, the sample pairs $(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_N, y_N)$ can be regarded as independently and identically distributed (i.i.d.). For simplicity, we introduce two

random variables X and Y for these samples. It allows us to model the joint distribution $f_{X,Y}(\boldsymbol{x}, y)$ via $\mathcal{M}$ with estimated parameters $\hat{\Theta}$, i.e., $f_{X,Y}(\boldsymbol{x}, y; \mathcal{M}, \hat{\Theta})$, and derive $f_{Y|X}(y|\boldsymbol{x})$ via the conditional probability formula. It is described as

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

With the estimated joint distribution $f_{X,Y}(\boldsymbol{x}, y; \mathcal{M}, \hat{\Theta})$ at hand, at any time t, given contextual information $\boldsymbol{x}_t$, one can issue the forecast $\hat{f}_{Y|X}(y_t|\boldsymbol{x}_t; \mathcal{M}, \hat{\Theta})$ 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 $\boldsymbol{x}_t$ and y_t as $\boldsymbol{z}_t$, i.e., $\boldsymbol{z}_t = [\boldsymbol{x}_t^\top, y_t]^\top$. Accordingly, the dataset is concatenated as the matrix $\boldsymbol{Z}$ of shape $N \times (pk + 1)$, i.e.,

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

We refer to the i-th row, j-th column, and (i, j)-th entry of $\boldsymbol{Z}$ as $\boldsymbol{z}_i$, $\boldsymbol{Z}_j$, 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., $Z = (Z_1, Z_2, \cdots, Z_{pk+1})$. Then, the distribution of Z is modeled by $f_Z(\boldsymbol{z}; \mathcal{M}, \hat{\Theta})$. In particular, let Z_{-j} denote the collection of random variables in Z except Z_j, i.e., $Z_{-j} = (Z_1, \cdots, Z_{j-1}, Z_{j+1}, \cdots, Z_{pk+1})$. Accordingly, let $\boldsymbol{z}_{-j}$ denote the realization of Z_{-j}.

We assume values are missing at random, which means that missingness is not dependent on values of missing entries. Obviously, missing values are likely to occur in every element of $\boldsymbol{z}_t$. Let us introduce a vector $\boldsymbol{m}_t$ to indicate the missingness of $\boldsymbol{z}_t$. 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 $\boldsymbol{M}$ indicates the missingness of $\boldsymbol{Z}$. Let $\mathcal{J}_{\boldsymbol{z}_t,M}$ denote the indices of missingness of $\boldsymbol{z}_t$, i.e., $\mathcal{J}_{\boldsymbol{z}_t,M} = \{j \mid m_{t,j} = 1\}$, and $\mathcal{J}_{\boldsymbol{z}_t,O}$ denote the indices of observations, i.e., $\mathcal{J}_{\boldsymbol{z}_t,O} = \{j \mid m_{t,j} = 0\}$. Therefore, the observed and missing parts of $\boldsymbol{z}_t$ are represented by $\boldsymbol{z}_t[\mathcal{J}_{\boldsymbol{z}_t,O}]$ and $\boldsymbol{z}_t[\mathcal{J}_{\boldsymbol{z}_t,M}]$, which are written as $\boldsymbol{z}_t^{obs}$ and $\boldsymbol{z}_t^{mis}$ for simplicity. The corresponding random variables for $\boldsymbol{z}_t^{obs}$ and $\boldsymbol{z}_t^{mis}$ are denoted as Z^{obs} and Z^{mis}. When y_t is missing, $\boldsymbol{z}_t^{mis} = [\boldsymbol{x}_t^{mis\top}, y_t]^\top$ where $\boldsymbol{x}_t^{mis}$ is the missing part of $\boldsymbol{x}_t$. The corresponding random variables for $\boldsymbol{x}_t^{mis}$ are denoted as X^{mis}. For example, Figure 1 presents the matrix $\boldsymbol{Z} = [z_{i,j}]_{4\times4}$, where blue blocks indicate observations and yellow blocks indicate missing values. As shown, the first row of $\boldsymbol{Z}$ is denoted as $\boldsymbol{z}_1$, the second entry of which, i.e., $z_{1,2}$ is missing. Then, the indices of missing values and observations of $\boldsymbol{z}_1$ are $\mathcal{J}_{\boldsymbol{z}_1,M} = \{2\}$ and $\mathcal{J}_{\boldsymbol{z}_1,O} = \{1, 3, 4\}$. Accordingly, we have $\boldsymbol{z}_1^{obs} = [z_{1,1}, z_{1,3}, z_{1,4}]^\top$, $\boldsymbol{z}_1^{mis} = [z_{1,2}]$. The corresponding random variables for $\boldsymbol{z}_1^{obs}$ and $\boldsymbol{z}_1^{mis}$ are denoted as $Z^{obs} = (Z_1, Z_3, Z_4)$ and $Z^{mis} = Z_2$. Also, let $\mathcal{I}_{\boldsymbol{Z}_j,M}$ denote the indices of missing values in $\boldsymbol{Z}_j$, i.e., $\mathcal{I}_{\boldsymbol{Z}_j,M} = \{i \mid m_{i,j} = 1\}$, and $\mathcal{I}_{\boldsymbol{Z}_j,O}$ denote the

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

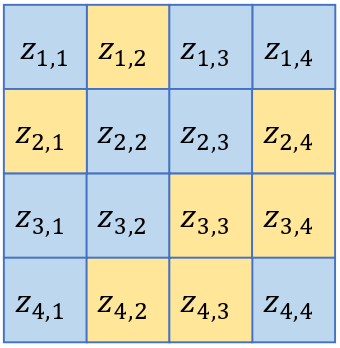


Figure 1: Illustration of a dataset $\boldsymbol{Z}$. Here we take $p = 1$, $k = 3$, $h = 1$ as an example. Blue blocks indicate observations, whereas yellow blocks indicate missing values.

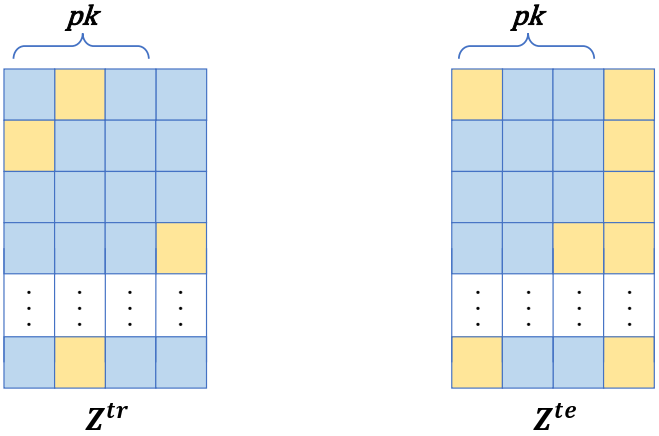


Figure 2: Illustration of training and test datasets. Here we take $p = 1$, $k = 3$, $h = 1$ as an example. Blue blocks indicate observations, whereas yellow blocks indicate missing values.

Therefore, at the model estimation phase, we concatenate features and targets to form a training dataset $\boldsymbol{Z}^{tr}$ with some missing values, based on which an imputation model $\mathcal{M}$ is trained. At the operational forecasting phase, the input vector $\boldsymbol{x}_t$ at time t is available, part of which may be missing, and we focus on target y_t. Together, they form $\boldsymbol{z}_t = [\boldsymbol{x}_t^\top, y_t]^\top$. Then $\boldsymbol{z}_t$ 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 $\boldsymbol{z}_t$. It is described as

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

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

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

Therefore, to learn the parameters Θ, it is required to maximize the likelihood of observations only, i.e., $f_{Z^{obs}}(\boldsymbol{z}_i^{obs}; \mathcal{M}, \Theta)$. 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 $f_Z(\boldsymbol{z}; \mathcal{M}, \hat{\Theta})$ with estimated parameters $\hat{\Theta}$ at hand, and show how to issue forecasts at the operational forecasting stage. If $\boldsymbol{x}_t$ is fully observed, then $\boldsymbol{x}_t$ is the observed part of $\boldsymbol{z}_t$, i.e., $\boldsymbol{z}_t^{obs} = \boldsymbol{x}_t$, whereas the missing part of $\boldsymbol{z}_t$ is y_t. The forecast for $\boldsymbol{x}_t$ can be expressed as

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

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

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

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

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

3. Forecasting with missing values via FCS

In this section, we develop a forecasting approach based on the proposed UI strategy. Concretely, we employ the FCS framework, which is based on Gibbs sampling. In what follows, we will introduce the FCS framework and its main components.

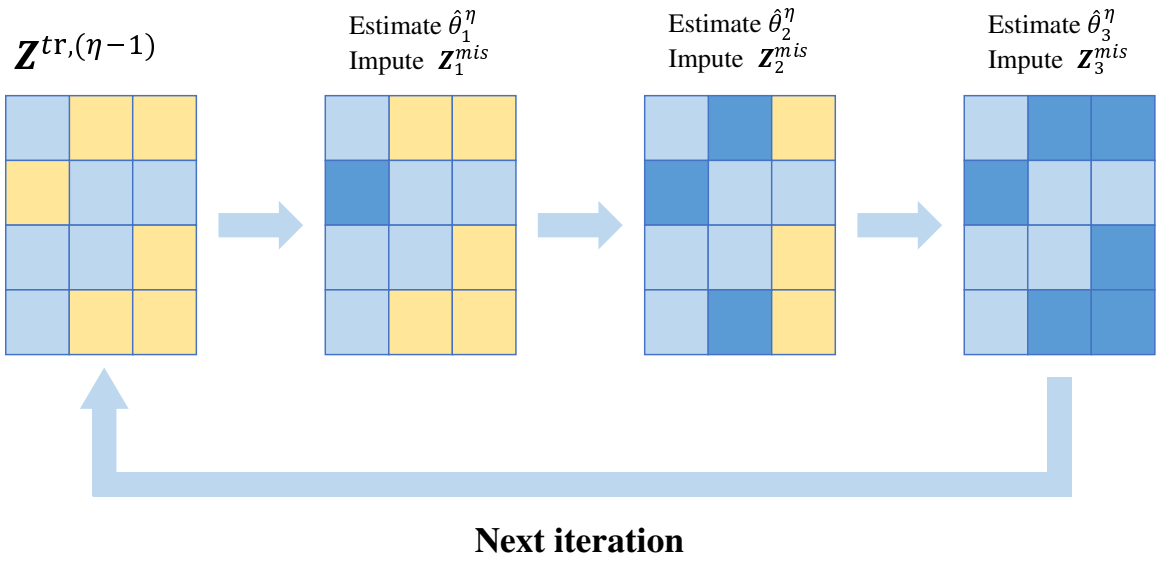


Figure 3: Illustration of the η-th iteration at the training stage. Light blue blocks indicate observations, yellow blocks indicate missing values, and dark blue blocks indicate imputation.

3.1. Fully conditional specification method

Instead of defining a multivariate distribution $f_Z(\boldsymbol{z}; \mathcal{M}, \hat{\Theta})$ 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 $f_{Z_j|Z_{-j}}(z_j|\boldsymbol{z}_{-j}; g_j, \hat{\theta}_j)$. 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 $\boldsymbol{Z}^{tr}$. For simplicity of notations, we still use $\boldsymbol{Z}$ in what follows to show how to estimate the parameters. Intuitively, before estimating $\hat{\theta}_j$, one needs to impute the missing values of $\boldsymbol{Z}_{-j}$. Then, parameters are estimated based on the imputed $\boldsymbol{Z}_{-j}$ and $\boldsymbol{Z}_j^{obs}$. With the estimated conditional distribution $f_{Z_j|Z_{-j}}(z_j|\boldsymbol{z}_{-j}; g_j, \hat{\theta}_j)$, one can impute $\boldsymbol{Z}_j^{mis}$ based on the corresponding conditionals in $\boldsymbol{Z}_{-j}$. That is, both the estimation of $\hat{\theta}_j$ and the imputation of $\boldsymbol{Z}_j^{mis}$ are based on the imputed $\boldsymbol{Z}_{-j}$. Obviously, the imputation of any column of $\boldsymbol{Z}_{-j}$, for instance $\boldsymbol{Z}_q$, relies on its conditional distribution $f_{Z_q|Z_{-q}}(z_q|\boldsymbol{z}_{-q}; g_q, \hat{\theta}_q)$, which requires $\boldsymbol{Z}_j^{mis}$ to be imputed. In other words, the estimation of $\hat{\theta}_j$ and the imputation of $\boldsymbol{Z}_{-j}$ are coupled with each other. If one performs the parameter estimation and imputation sequentially for $j = 1, 2, \cdots, pk+1$, the estimation of $\hat{\theta}_j$ can only use initial imputation of $\boldsymbol{Z}_{j+1}^{mis}, \cdots, \boldsymbol{Z}_{pk+1}^{mis}$. The updated imputation of $\boldsymbol{Z}_{j+1}^{mis}, \cdots, \boldsymbol{Z}_{pk+1}^{mis}$ given by their estimated conditional distributions cannot be used for the estimation of $\hat{\theta}_j$. Therefore, we perform the imputation of $\boldsymbol{Z}_j$ and the estimation of $\hat{\theta}_j$ in an iterative manner. Then, at next iteration, the updated imputation of $\boldsymbol{Z}_{j+1}^{mis}, \cdots, \boldsymbol{Z}_{pk+1}^{mis}$ 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 $\boldsymbol{Z}_j^{(\eta)}$. At the $\eta+1$-th iteration, $\boldsymbol{Z}_{j+1}^{(\eta)}, \cdots, \boldsymbol{Z}_{pk+1}^{(\eta)}$ 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 $\boldsymbol{Z}_j$ becomes complete and is written as $\boldsymbol{Z}_j^{(0)}$. 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 $\boldsymbol{Z}_1^{(\eta)},\cdots,\boldsymbol{Z}_{j-1}^{(\eta)},\boldsymbol{Z}_{j+1}^{(\eta-1)},\cdots,\boldsymbol{Z}_{pk+1}^{(\eta-1)}$ at hand, which are written compactly as $\boldsymbol{Z}_{-j}^{(\eta)}$ in the form of a matrix, i.e.,

$$\boldsymbol{Z}_{-j}^{(\eta)} = [\boldsymbol{Z}_1^{(\eta)},\cdots,\boldsymbol{Z}_{j-1}^{(\eta)},\boldsymbol{Z}_{j+1}^{(\eta-1)},\cdots,\boldsymbol{Z}_{pk+1}^{(\eta-1)}]. \tag{11}$$

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

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

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

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

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

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

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

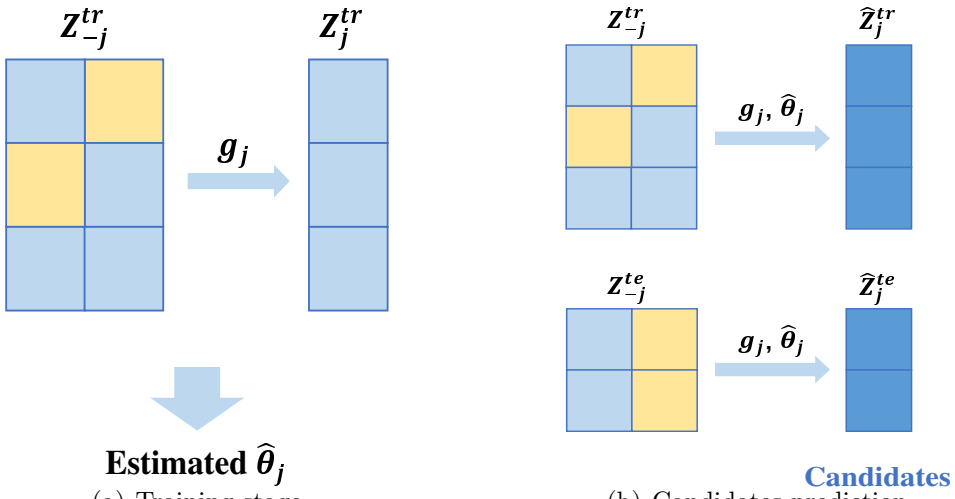
$$\boldsymbol{Z}_j^{(\eta)} = [z_{1,j}^{(\eta)},\cdots,z_{N,j}^{(\eta)}]^\top. \tag{15}$$

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

3.2. Predictive mean matching

In this paper, $f_{Z_j|Z_{-j}}(z_j|\boldsymbol{z}_{-j}; \hat{\theta}_j, g_j)$ 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 $\boldsymbol{z}_{-j}$ to z_j, i.e.,

$$z_j = g_j(\boldsymbol{z}_{-j}; \theta_j) + \epsilon_j, \tag{16}$$



(a) Training stage (b) Candidates prediction

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

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 $\boldsymbol{Z}^{tr}_{-j}$ and $\boldsymbol{Z}^{tr}_{j}$. With the estimated model, we respectively predict targets for $\boldsymbol{Z}^{tr}_{-j}$ and $\boldsymbol{Z}^{te}_{-j}$, which are called candidates and written as $\hat{\boldsymbol{Z}}^{tr}_{j}$ and $\hat{\boldsymbol{Z}}^{te}_{j}$. Then, for each entry of $\hat{\boldsymbol{Z}}^{te}_{j}$, we form a set of its d closest candidates in $\hat{\boldsymbol{Z}}^{tr}_{j}$, from which we perform random sampling to obtain imputations.

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

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

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

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

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

$$\hat{\boldsymbol{Z}}^{(\eta)}_{j} = [\hat{z}^{(\eta)}_{1,j}, \hat{z}^{(\eta)}_{2,j}, \cdots, \hat{z}^{(\eta)}_{N,j}]^{\top}. \tag{19}$$

To impute $\boldsymbol{Z}^{mis}_{j}$, 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 $\hat{z}^{(\eta)}_{i_m,j}$. Then we find d nearest candidates from $\hat{\boldsymbol{Z}}^{(\eta)}_{j}[\mathcal{I}_{j,obs}]$ for which $|\hat{z}^{(\eta)}_{i_m,j} - \hat{z}^{(\eta)}_{i,j}|, i_m \in \mathcal{I}_{j,mis}, i \in \mathcal{I}_{j,obs}$ is minimal. Suppose the d candidates are

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

which can be written in the form of a set as $\mathcal{C}_{i,j}$, i.e., $\mathcal{C}_{i,j} = \{\hat{z}^{(\eta)}_{i_1,j}, \hat{z}^{(\eta)}_{i_2,j}, \cdots, \hat{z}^{(\eta)}_{i_d,j}\}$. 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 $z^{(\eta)}_{i_m,j}$, i.e.,

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

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 $\hat{\boldsymbol{Z}}_j$, which are prepared for the use of sampling at the operational forecasting stage.

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 $\boldsymbol{z}_{-j}$ to z_j is described as

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

where $g_{j,b}(\boldsymbol{z}_{-j})$ 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., $R_1, R_2, \cdots, R_M$. And we model the target as a constant c_m in each region. The regression function is described as

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

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.,

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

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

$$R_1(a,s) = \{\boldsymbol{z}_{-j} | z_a \le s\},\ R_2(a,s) = \{\boldsymbol{z}_{-j} | z_a > s\}. \tag{24}$$

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

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

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

3.4. Forecasting Stage

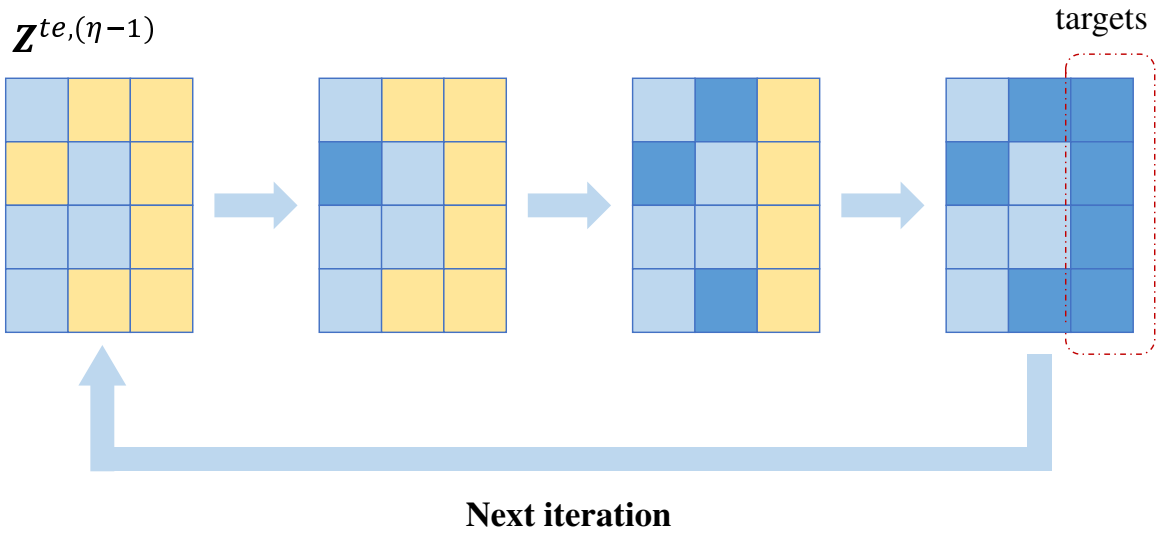


Figure 5: Illustration of the η-th iteration at the operational forecasting stage. Light blue blocks indicate observations, yellow blocks indicate missing values, and dark blue blocks indicate imputation (forecasting).

After training the imputation model, we obtain a collection of estimated random forests $\{g_j\}$ with parameters $\{\hat{\theta}_j\}$ and candidates $\{\hat{\boldsymbol{Z}}_j\}$. At the operational forecasting stage, we feed sample $\boldsymbol{z}_t = [\boldsymbol{x}_t^\top, y_t]^\top$ (y_t is missing by default) into the estimated imputation model, and iteratively impute each missing value in $\boldsymbol{z}_t$ 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 $\boldsymbol{z}_t$ are obtained, which are written as

$$\tilde{\boldsymbol{z}}_t^1, \tilde{\boldsymbol{z}}_t^2, \cdots, \tilde{\boldsymbol{z}}_t^L.$$

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

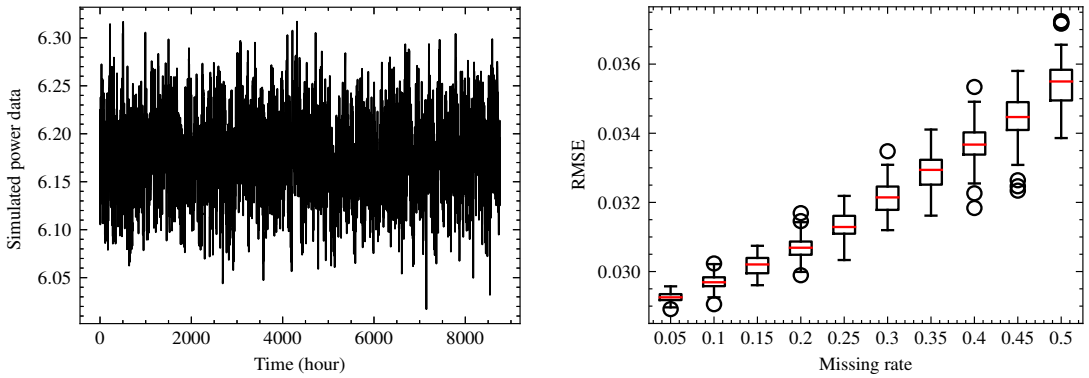
$$\tilde{y}_t^i = \tilde{z}_{t,pk+1}^i, \quad i = 1, \cdots, L. \tag{26}$$

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 $f_{Y_{n,t+h}|t}(y|\boldsymbol{x}_t; \mathcal{M}, \hat{\Theta})$. Thus we rewrite $\tilde{y}_t^i$ as $\tilde{y}_{n,t+h|t}^i$, all of which form a set, i.e.,

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

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

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



(a) Simulated data (b) Results

Figure 6: (a) Simulated time series based on the AR process; (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 (base on Monte-Carlo with 100 replications).

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

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

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.

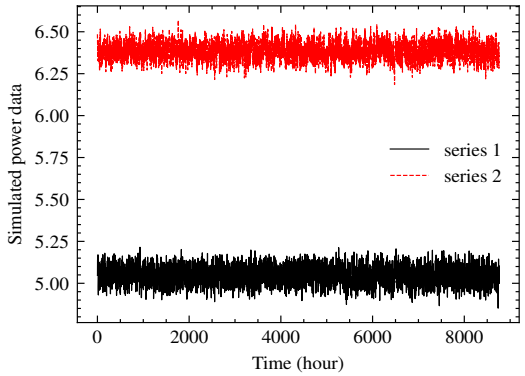
4.1. AR process

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

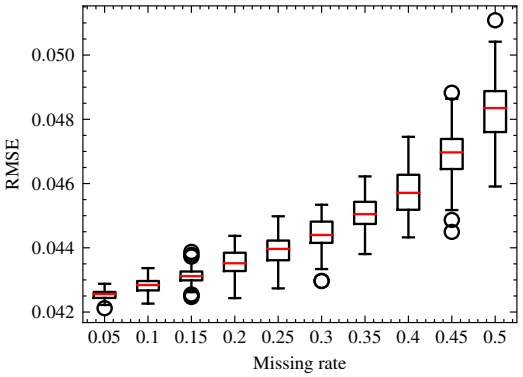
$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \epsilon_t,$$

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 $\boldsymbol{x}_t$ have 2 dimensions, and target y_t has

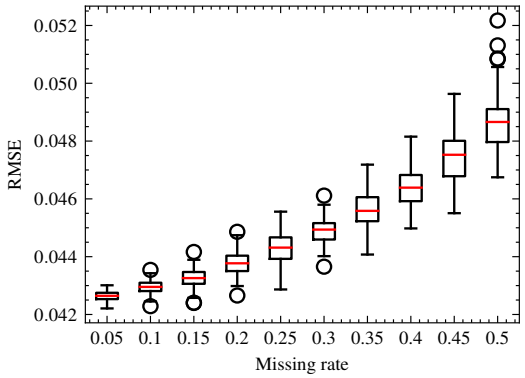
1 dimension. Specifically, the imputation model is trained by 10 iterations, as suggested by Van Buuren et al. (2006). At the operational forecasting stage, 200 realizations are generated in parallel, based on which point forecasts are derived according to (26) and (27). The RMSE values with respect 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. 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.



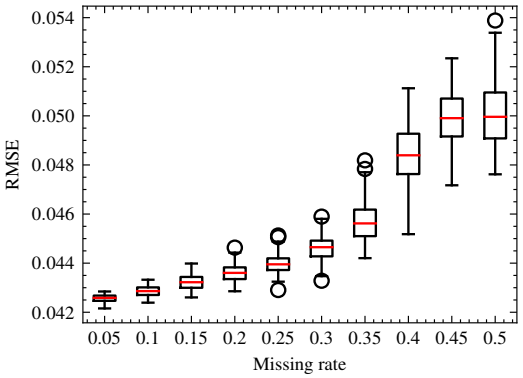
(a) Simulated data



(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).

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 via

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

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

$$Y_{1,t} = \alpha_{1,1} Y_{A,t} + \alpha_{1,2} Y_{B,t} + \epsilon_{1,t},$$
$$Y_{2,t} = \alpha_{2,1} Y_{A,t} + \alpha_{2,2} Y_{B,t} + \epsilon_{2,t},$$

where $\boldsymbol{\alpha}_1^\top = [0.9,\ 0.1]^\top$, $\boldsymbol{\alpha}_2^\top = [0.3,\ 0.7]^\top$, $\epsilon_{1,t} \sim \mathcal{N}(0, 0.01)$, and $\epsilon_{2,t} \sim \mathcal{N}(0, 0.01)$. 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 $\boldsymbol{x}_t$ 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 tha 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 to illustrate the applicability of our approach, we further validate it based on real-world data from the USA and Denmark. It still contains a training stage and a genuine out-of-sample forecast verification period. Both point and probabilistic forecasting are considered. And, we apply this approach to both single wind farm and multiple wind farms scenario where one can utilize data from nearby wind farms to improve the quality of forecasts. We note that the goal of this case study is not to find a state-of-the-art method, but to show the impacts of missing values on forecasting and the effectiveness of the proposed approach. In what follows, first, we describe the data sets, assessment metrics, and benchmark models. Then validation of the proposed approach is presented.

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. Data from Denmark are gathered for an offshore wind farm, which are real wind power generation values and show missingness in practice.

5.1.1. Dataset from the USA (South Carolina)

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. Besides, we simulate different missingness for each wind farm.

5.1.2. Dataset from Denmark (Anholt)

The dataset comes from real wind power generation values at the Anholt offshore wind farm located in Denmark, which aggregates power generation values from 110 wind turbines. It contains data from the 1st of July 2013 to the 31st of August 2014, with the resolution of 10 minutes. Missing values occur in practice due to communication or sensor errors and account for 9.5% of data.

5.2. Experimental setups

Before reporting results, we describe setups, assessment metrics, and benchmark models in this subsection.

5.2.1. Setups

We perform both point and probabilistic forecasting in three cases, which are:

(1) Forecasting at a single site based on the data from the USA. The missing rate is set as 20% here. Particularly, we consider different lead times in the point forecasting setting.

(2) Forecasting of aggregated wind power generation values based on the data from Denmark. Lead time is set as 1.

(3) Forecasting at a chosen site by using features from nearby sites based on the data from the USA. Lead time is set as 1.

In all cases, values at previous time steps are used as input features. As feature selection is not the focus of this paper, we test different lengths in a preliminary study and empirically set the order of autoregressive models to 6 lags. And, generalized logit-normal transform (Pinson, 2012) can be further employed as a pre-processing stage to accommodate the boundary characteristics of wind power generation values (e.g., nonlinearity and variance of residuals conditional to the mean level) .

5.2.2. Assessment metrics

The quality of point forecasts is evaluated with an RMSE criterion, whereas the quality of probabilistic forecasts is assessed by using the Continuous Ranked Probability Score (CRPS). Denote the cumulative density function of wind power Y_t as F_t at time t. Then, CRPS for the estimated F_t is defined as

$$\mathrm{CRPS}(F_t, y_t) = \int_y (F(y) - \mathbb{1}(y - y_t))^2 dy, \tag{29}$$

where $\mathbb{1}(\cdot)$ is a unit step function, which can be regarded as the empirical cumulative density function of the observation y_t. Here we report the average over all forecast-verification pairs, i.e.,

$$\mathrm{CRPS} = \frac{1}{|\mathcal{I}_{y,obs}|} \sum_{t \in \mathcal{I}_{y,obs}} \mathrm{CRPS}(F_t, y_t). \tag{30}$$

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. And, 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.

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 UI approach is still implemented with the copula-based model.

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

In this case, besides simulating the missingness, we consider a reference model that is implemented by a random forest and trained based on the complete dataset to show the influence of missingness. The results of point forecasting in the term of RMSE are presented in Table 1. Not surprisingly, the RMSE increases with the lead time. It can be clearly seen that the quality of forecasts in the presence of missing values are worse than those

Table 1: RMSE with different lead times in Case 1 (percentage of normalized capacity).

Lead Time	Persistence	RF-M[1]	RF-R[2]	Copula	FCS	RF-complete[3]
1	16.8	20.1	16.6	17.1	15.9	14.6
2	19.1	23.2	20.4	19.1	18.2	16.9
3	21.1	25.8	23.9	20.9	19.8	18.6
6	26.1	30.6	31.2	25.3	23.5	22.4

[1] ITP strategy where random forests and mean imputation are employed.
[2] ITP strategy where random forests and regression-based imputation are employed.
[3] A reference model where dataset is complete.

on the condition of complete data, since missing values affect both model estimation and operational forecasting stages. The persistence method still serves as a good benchmark, as it is easy to be implemented and its quality is not too bad even compared to the reference model. As shown, the two methods using ITP strategy, i.e., RF-M and RF-R perform worse than persistence method, owing to errors introduced by the pre-processing procedure. Given training datasets $\boldsymbol{X}^{tr}$ and $\boldsymbol{Y}^{tr}$, 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 $\boldsymbol{X}^{tr,P}$ and $\boldsymbol{Y}^{tr,P}$. However, the imputed datasets are usually deviated from the real complete datasets. Then, the model estimated based on $\boldsymbol{X}^{tr,C}$ and $\boldsymbol{Y}^{tr,C}$, 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.

The used copula-based method and FCS method 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. Indeed, the copula and FCS based models outperform RF-M and RF-R, as revealed in Table 1. Although copula method allows to characterize several kinds of distributions, it requires 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 is free of such an assumption, and therefore has a better performance than the copula-based model. However, the improvement in performance is gained at the cost of increased complexity, which will be further discussed in following subsections.

Next, we move to the results of probabilistic forecasting, the CRPS values of which are presented in Table 2. Here, missing values have nearly no influence on the performance of climatology, since the climatology method characterizes uncertainty via the empirical distribution of all historical values. Comparing the Gaussian-M and Gaussian-R, we know that a better imputation method is still preferred by the ITP strategy in the context of probabilistic forecasting. Both the Gaussian-R and QR-R use the regression-based imputation as preprocessing procedure. But they differ in backbone models – Gaussian-R relies on the

Table 2: CRPS in Case 1 (percentage of normalized capacity).

Climatology	Gaussian-M[1]	Gaussian-R[2]	QR-R[3]	Copula	FCS	QR-complete[4]
18.6	12.4	9.4	9.2	11.3	7.2	7.4

[1] ITP strategy where parametric model with Gaussian assumption and mean imputation are employed.
[2] ITP strategy where parametric model with Gaussian assumption and regression-based imputation are employed.
[3] ITP strategy where where quantile regression and regression-based imputation are employed.
[4] A reference QR model where dataset is complete.

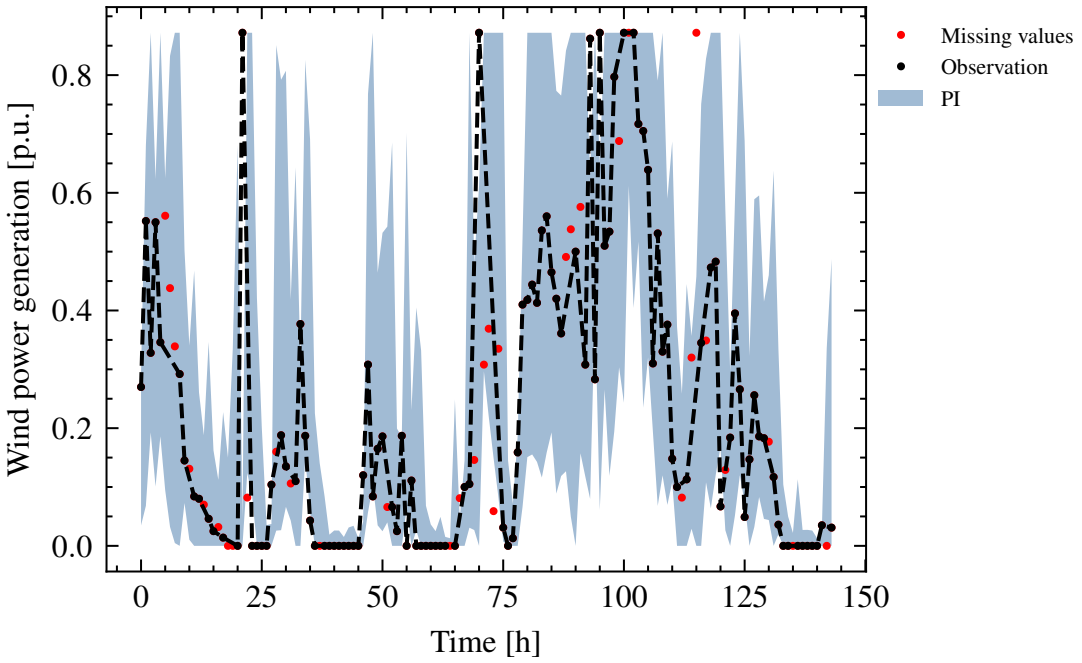


Figure 8: Illustration of 90% prediction intervals over a period of 6 days, as issued by the FCS-based model.

Gaussian distributional assumption, whereas QR-R is distribution-free. Their performance is comparable in this case, which is different from the situation on complete datasets where QR is always superior, revealing the impacts of missing values. Obviously, one needs to estimate two shape parameters of Gaussian distribution in Gaussian-R, but the parameters of several quantile functions in QR-R. The parallel model estimation of QR-R may introduce more errors to the ultimate estimated distribution. Therefore, results are governed by both the selection of models and the influence of missing values on model estimation. Indeed, the relative performance of Gaussian-R and QR-R may depend on missingness, which will be further discussed in next case. The FCS-based model outperforms Gaussian-R and QR-R, whereas the performance of copula-based method is worse than those of Gaussian-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

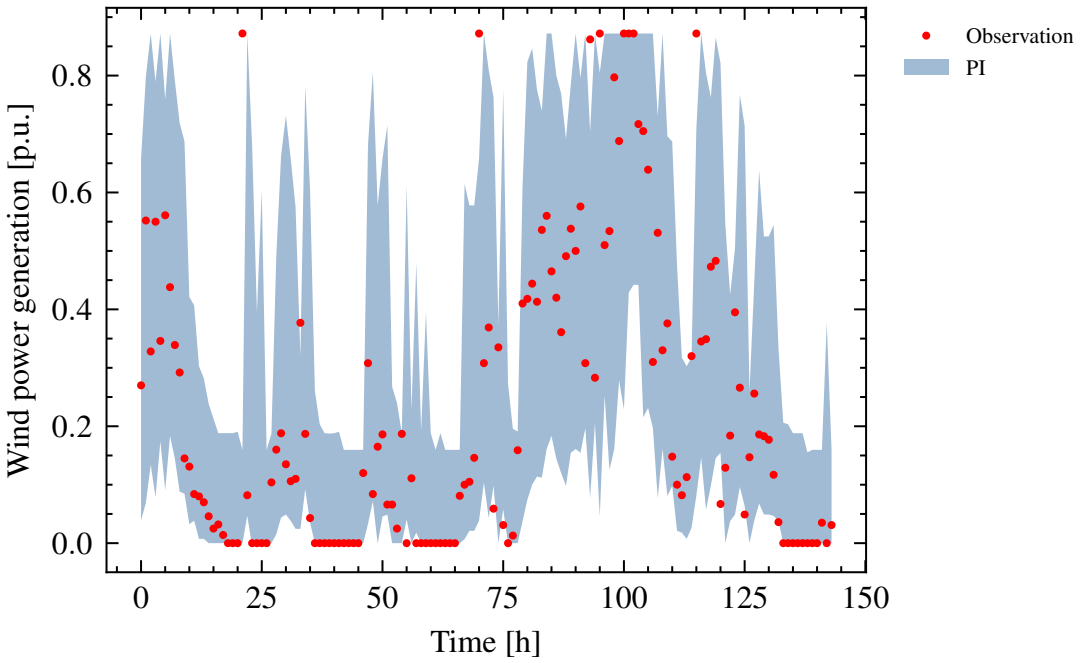


Figure 9: 90% prediction intervals over a period of 6 days, as issued by the reference model.

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 model and reference model respectively in Figure 8 and Figure 9 for illustration. As shown, although FCS-based model encounters missing values at model estimation and operational forecasting stages, its PIs are similar to those of reference model. Specially, in some time interval such as the interval from 35-h to 45-h, the PIs issued by FCS-based model are sharper than those of reference model.

5.3.2. Case 2

In this case, we perform experiments based on data from Denmark, which contain real missing values; thus we cannot set a reference model like in Case 1 now. Different from the simulated missingness in Case 1 (referred to as sporadic missingness), missing values occur in blocks (referred to as block missingness) here. In other words, missingness sporadically occurs at almost every sample in Case 1, and therefore the values of a sample are not entirely missing. By contrast, in this case we have several samples whose values are entirely missing or observed. The RMSE values are presented in Table 3. As with the Case 1, RF-M is worse than RF-R, which suggests that by using the ITP approach, advanced imputation is still preferred on the condition of block missingness. However, unlike in Case 1 where the UI approach overwhelmingly outperforms the ITP approach, the copula-based model has the worst performance here. Indeed, the estimation of copula-based model is based on 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. The FCS-based model still obtains the lowest RMSE, but the RMSE is similar to that of RF-R.

Table 3: RMSE in Case 2 (percentage of normalized capacity).

Persistence	RF-M	RF-R	Copula	FCS
4.8	5.0	4.2	6.1	4.2

The CRPS values are presented in Table 4, where a persistence model is employed which estimates the empirical distribution via recent 100 observations. Not surprisingly, the performance of Gaussian-R is still superior to that of Gaussian-M. However, different from Case 1, QR-R performs worse than Gaussian-R, which might be attributed to the block missingness. In other words, the imputation for samples whose values are entirely missing may jeopardize the estimation of QR models. By contrast, the estimation of model with a Gaussian distributional assumption is more robust, as samples deviate from the underlying distribution will have less weights in the computation of likelihood. FCS-based model has the best performance among all models, which suggests that FCS method is effective and robust to different kinds of missingness.

Table 4: CRPS in Case 2 (percentage of normalized capacity).

Climatology	Persistence	Gaussian-M	Gaussian-R	QR-R	Copula	FCS
23.2	11.8	1.8	1.7	2.6	5.4	1.4

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), based on data from the USA. 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. Specifically, we simulate different missing rates at two nearby wind farms, the RMSE values of which are shown in Table 5. Obviously, the accuracy of point forecasting is improved with the assistance of AFs, which is comparable to RF-complete in the Table 1. Furthermore, it can be seen that the benefit of AFs is robust, since the performance is relatively steady as the missing rate of AFs increases. It might be explained by that the key information for forecasting comes from the target wind farm itself. So, it will not make a big difference when few auxiliary features are missing. The results of probabilistic forecasting is shown in Table 5, which also suggests that AFs provide extra information and thus contribute to improving probabilistic forecasts.

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

Table 5: RMSE and CRPS in Case 3 (percentage of normalized capacity).

	No AFs	AFs	AFs 5% m[1]	AFs 10% m[2]	AFs 20% m[3]
RMSE	15.9	14.4	14.7	14.6	14.7
CRPS	7.2	6.1	6.1	6.2	6.2

[1] 5% of AFs are missing.
[2] 10% of AFs are missing.
[3] 20% of AFs are missing.

Table 6: Training time and operational time in Case 1.

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

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 6 for illustration. Therefore, it is required to find computationally efficient methods to implement the proposed approach.

6. Conclusions

It is natural 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 (as exemplified by the Danish real-world example, with nearly 10% of data missing). 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) and real operational data (from an offshore wind farm in Denmark) 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. 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 computation costs for the introduced FCS method are high, and growing significantly as the dimension increases. Therefore, more efficient methods are still in need. Besides, as the proposed approach relies on a stationarity assumption for the underlying stochastic process, emphasis should be placed on relaxing that assumption in the future in order to deal with nonstationary environments, e.g., with online learning. The situation where observations are missing not at random is also worthy to be further explored.

References

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.

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.
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.
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 rkhss, 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.
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., 2021. Continuous and distribution-free probabilistic wind power forecasting. TechRxiv. Preprint. techrxiv.16866280.v1 .
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.