Mape_Maker: A Scenario Creator

Guillaume Goujard1 · Jean-Paul Watson2 · David L. Woodruff3

Received: 10 October 2019 / Accepted: 19 September 2020 / Published online: 29 September 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract
We describe algorithms for creating probabilistic scenarios for renewables power production. Our approach allows for tailoring of forecast uncertainty, such that scenarios can be constructed to capture the situation where the underlying forecast methodology is more (or less) accurate than it has been historically. Such scenarios can be used in studies that extend into the future and may need to consider the possibility that forecast technology will improve. Our approach can also be used to generate alternative realizations of renewable energy production that are consistent with historical forecast accuracy, in effect serving as a method for creating families of realistic alternatives—which are often critical in simulation-based analysis methodologies. We illustrate the methods using real data for day-ahead wind forecasts.

Nomenclature

Observed Variables
\( x_t \) Timeseries of independent input data (e.g. actuals)
\( y_t \) Timeseries of dependent input data (e.g. forecasts)
\( \mathcal{X} \) Set of paired input data (actuals, forecasts) or (forecasts, actuals)
\( x_{SID}^t \) Timeseries of simulation input data
\( \mathcal{X}_{SID} \) Set of Simulation Input Data (SID) upon which the simulation is performed
\( \tilde{r} \) Target MARE (Mean Absolute Relative Error)

Random Variables
\( \tilde{\epsilon}_t \) Random vector of simulated errors
\( \tilde{y}_t \) Random vector of simulated values
\( \tilde{u}_t \) Random vector of uniform base process
Random variable of the error
\( \varepsilon \)
Random variable of the simulated error
\( \hat{\varepsilon} \)
Random variable of the input
\( X \)
Joint random variable: \( Z = (\varepsilon, X) \)
Marginal density function of the input data \( X \)
\( f_X \)
Marginal density function of the error random variable
\( f_\varepsilon \)
Conditional density function of the error given the input
\( f_{\varepsilon|X=x} \)
Cumulative distribution function of the error given the input
\( F_{\varepsilon|X=x} \)

**Estimation**

- \( a \) Percent of data used to estimate each conditional distribution
- \( I_x^a \) Interval of \( 2a \) fraction of data around \( x \) in \( X \)
- \( \bar{x}(x,a) \) Center of the interval \( I_x^a \)
- \( \text{cap} \) Capacity
- \( b(\cdot;\alpha,\beta,l,s) \) Density function of a beta for parameters: \( \alpha, \beta, l, s \)
- \( \hat{S}_x \) Set of estimated beta parameters of the conditional distributions \( \varepsilon \mid X = x \) over \( X \)
- \( \hat{m}(x) \) Expected value of the absolute estimated error given the input
- \( \hat{m}(x, \hat{r}, \omega) \) Expected value of the absolute simulated error given the input, a target MARE, and a weight function
- \( \omega_\chi(\cdot) \) Weight function over \( \chi \)
- \( \hat{F}_{\varepsilon|X=x} \) Cumulative distribution function of the estimated error given the input
- \( \hat{f}_{\varepsilon|X=x} \) Estimated conditional density function of the error given the input in \( X \)
- \( m_{\max}(x) \) Maximum value of the expected value of the absolute estimated error, given \( x \)
- \( r_{\hat{m}} \) Expected value of the mean absolute relative estimated error over \( X \)
- \( \hat{r} \) Mean absolute relative error over \( X \) under the estimated conditional distributions

**Simulation Distributions**

- \( \tilde{F}_{\varepsilon|X=x} \) Cumulative distribution function of the simulated error given the input
- \( \tilde{f}_{\varepsilon|X=x} \) Simulation conditional density function of the error given the input in \( X_{\text{SID}} \)
- \( \hat{S}_{x,m} \) Set of simulation beta parameters of the conditional distributions \( \varepsilon \mid X = x \) over the SID
- \( \omega_{\chi_{\text{SID}}}(\cdot) \) Weight function over \( X_{\text{SID}} \)
- \( P_{\text{SID}} \) Distribution plausibility score

**Base Process**

- \( \tilde{z}_t \) Random vector of base process
- \( \phi \) Cdf of the standard normal distribution
- \( (\tilde{z}_t)_{t \leq n} \) Estimated base process
Mape Maker: A Scenario Creator

\[(a_t, b_t)_{t \leq p}\]  ARMA parameters of the estimated base process

\[\sigma_\delta\]  Standard deviation of the error of the estimated base process

**Simulation**

\[d\]  Mean of the curvature of target input data

---

1 Introduction

Uncertainty associated with the forecasted production of renewable energy sources such as wind and solar mandates power systems analysis and management techniques that directly take stochastics into account. A growing literature describes methods for creating and evaluating probabilistic *scenarios*, which are forecasts of renewables power production with an attached probability. A representative sample of this literature can be found in [3, 7–9, 12, 16]. Here, we are interested in creating probabilistic scenarios for the situation when the underlying forecast methodology is modeled as being more (or less) accurate than it has been historically, e.g., to capture potential improvements in underlying numerical weather prediction models. Such scenarios can be used in studies that consider future power system conditions and configurations, and therefore may need to consider the possibility that forecast technology will improve. Our approach can also be used to construct alternative realizations of actual renewable energy production that are consistent with the accuracy of a particular forecasting methodology, in providing a method for creating families of realistic alternatives—which are often critical in simulation-based analysis methodologies. A general open-source software implementation of the methods described here—a package called *mape_maker*—is publicly available at https://github.com/mape-maker/mape-maker. While we focus our studies and exposition on renewables power production (specifically because the associated forecasting errors are significantly higher than for load), our methods and software operate on generic time series quantities of interest.

Forecasts of renewables power production are commonly available at periodic intervals, with each forecast specifying a time series for a future system operating horizon, e.g., day-ahead for each hour of the next 7 days. Corresponding time series for *ex post* production are available once the system operates through the forecasted operating horizon; we refer to the latter time series simply as *actuals*. Our methods and software are based on the availability of a historical (e.g., over the past year) set of such forecasts and associated actuals, which would generally be available to any system operator.

Given such a historical data set, we create a set of renewables power production scenarios that could reasonably correspond to a forecast (a current forecast or a forecast from the past). We refer to these scenarios as “actuals” to distinguish them from forecasts, although they are synthetic alternatives to the realized renewables power production time series. We can also create a set of forecasts that could reasonably correspond to a given time series of actuals. In other words, the process can be inverted. The correspondence between forecasts and actuals is based on analysis of historical forecast error distributions. Subsequently, the word “reasonably” is
replaced with mathematical criteria concerning the error distribution, temporal correlation, and in the case of forecasts, curvature.

As a preview of the output of this capability, consider Fig. 1. This figure provides a simple example where a set of 5 alternative synthetic scenarios for wind power production are constructed for 7 days in July of 2013 for the California Independent System Operator (CAISO) in the US. The results are based on day-ahead wind forecast error data obtained from CAISO for July 2013 through May 2015. The target fit error—specifically, the mean absolute percentage error or MAPE—is the quantity realized in the historical forecast error data. Because the scenarios are created for days in the past, we are able to show both the historical forecasts and realized actuals on the same plot as the synthetic actual scenarios.

Our method represents, to the best of our knowledge, a first-of-kind capability for constructing either synthetic forecasts or realizations of renewables power production that are consistent with arbitrary forecast precision. Such a capability enables a range of novel analyses for power systems operations models that have to date—while of significant interest and potential impact to both practitioners and researchers—not been possible. In general, we view our methods as having the potential to significantly enhance the utility of production cost models (PCMs), which are the key analytic tool used by industry for power systems planning analysis.

1.1 Related Literature

The present work builds on prior research efforts conducted by the authors to construct probabilistic scenarios for wind power production [14], solar power production [16], and energy demand [4]. These works in turn are either related to or build on approaches for constructing probabilistic scenarios (largely in renewables power production), either to provide situational awareness regarding uncertainty to power system operators or to serve as input to advanced stochastic power systems operations optimization models.

![Fig. 1 Illustration of 5 scenarios of wind production in CAISO for several days in July 2013, each representing a synthetic realization. The historical forecasts and realized actuals are also shown](image-url)
One widespread approach to constructing probabilistic wind power scenarios involves fitting models using historically observed wind power production characteristics. For example, Morales et al. propose a methodology based on a time series analysis of historical wind power production, while also maintaining spatial correlation across distinct wind farms [6]. Such approaches are limited, as they do not incorporate information concerning short-term forecasts of resource availability, which provide the best information about near-term conditions. Early attempts to construct probabilistic wind power production scenarios from forecasts were straightforward. For example, Wang et al. assume wind power production is normally distributed around a forecasted quantity [15]. Pinson et al. proposed a greatly improved method based on multivariate Gaussian random variables, estimated with the covariance matrix of prediction errors [10]. The methods outlined in [14, 16] are designed to improve upon the Pinson et al. method, focusing on developing non-parametric forecast error distributions and controlled methods for generating low-probability scenarios. In [13], a time series based auto regressive moving average model is simulated to be transformed into actual wind speeds by distribution transformation. The wind speed scenarios are converted to wind power scenarios using a power curve. Unlike many methods and ours, their method does not consider an external forecast.

Associated with the above body of literature is research devoted to developing methods and metrics to assess the quality of the resulting scenarios. The vast majority of these approaches strictly consider static attributes of scenarios, in contrast to evaluating the scenarios in terms of their benefit in a decision context (e.g., power system operations). Well-known and widely studied examples include the energy score, the Brier Score, and Minimum Spanning Tree (MST) rank histograms [9]. More recent work in this area, intended to improve discrimination ability—such as the Variogram score, is reported in [11, 12]. In this paper, we introduce new criteria for scenarios and evaluate our methods using them.

Existing research on constructing probabilistic scenarios for renewables power production does not consider the ability to either (1) vary forecast errors in a controlled manner or (2) rigorously generate synthetic realizations that are consistent with a given forecast error target. There is also no literature concerned with creating forecast scenarios that plausibly correspond to observed power. This paper aims to fill these gaps.

1.2 Measures of Forecast Error

Let \((x_i)_t \in \mathbb{R}^n\) and \((y_i)_t \in \mathbb{R}^n\) denote two time series. For simplicity, we subsequently refer to these time series as \(x_t\) and \(y_t\). We then define the following functions:

\[
RE : \mathbb{R}^+ \times \mathbb{R} \rightarrow \mathbb{R} \quad (\text{Relative Error})
\]

\[
x, y \mapsto \frac{y - x}{x}
\]

\[
MARE : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+ \quad (\text{Mean Absolute Relative Error})
\]

\[
x_t, y_t \mapsto \sum_{i=1}^{n} \frac{|RE(x_i, y_i)|}{n}
\]

The MAPE (Mean Absolute Percentage Error) is simply the MARE (Mean Absolute Relative Error) given as a percentage. Our software library communicates with users
in terms of MAPE, but in our discussions here it is convenient to use MARE and sometimes MAE (Mean Absolute Error) variants.

While MAPE is a very popular way of characterizing forecast accuracy for renewables production, it is well-known to have a number of undesirable properties (see, e.g., [5]). One undesirable property is that \( x \) values of zero must be ignored in the calculation. We have organized our methods in such a way as to avoid division by zero. Most of the development here is based on converting the MAPE target to an absolute error conditional on the value of \( x \), so it would be a relatively straightforward extension to convert our algorithms to use some measure of accuracy other than the MAPE.

1.3 Notation Scheme

We use \( x_t \) as the time series of input data that will be treated as the independent and \( y_t \) as the time series of input data that will be modeled as dependent. These names make sense as our goal is ultimately to obtain a simulation time series \( \tilde{y}_t \), given an input time series \( x_{SID}^t \). Note that which of these pairs is the forecast and which is the actual depends on the user’s choice, i.e., what is being simulated.

Denote \( \mathcal{X} = \{(x_{i,t}, y_{i,t}), 1 \leq i \leq n\} \), the set of each paired occurrence. We use \( x_{SID}^t \), \( \mathcal{X}_{SID} \) (resp.), to denote the Simulation Input Data (SID) time series, set (resp.), upon which the simulation of \( \tilde{y} \) will be performed.

While the time series notation \( (x_t, y_t, x_{SID}^t) \) is going to come in handy when discussing the base process procedure (see Sect. 4), the set notation \( (\mathcal{X}, \mathcal{X}_{SID}) \) is going to be useful to discuss about the conditional distributions and obtain useful guarantees on the mean absolute relative error (MARE) of our simulations (see Sect. 2.3).

We use bold upper case font to denote random variables and bold lower case font to denote random vectors. We will focus on generating a plausible distribution of simulated errors \( \tilde{\epsilon}_t \) in the following. The title of the paper and the name of our software library derives from the requirement that simulated values \( \tilde{y}_t = x_{SID}^t + \tilde{\epsilon}_t \) must result in a MARE close enough to the target MARE. We formalize this constraint as

\[
\mathbb{E}[MARE(x_{SID}^t, x_{SID}^t + \tilde{\epsilon}_t)] = \tilde{r},
\]

where \( \tilde{r} \) is the target MARE, the target MAPE divided by 100%.

1.4 Plausibility Objectives

A main theme underlying this work that we will use to justify some of our design choices involves what we refer to as plausibility objectives. For any requested MARE, the distribution of errors computed should be as close as possible to the original error distribution while satisfying the target MARE. If a user were to select the estimated MARE as the requested one, one would naturally expect the distribution of errors drawn from the simulated distributions to be somehow “close” to the estimated distribution. For example, if the system of forecasts is producing a wide range of errors at very low forecasted power output, then even if the forecast technology is improving, one would expect it to still produce a relatively wider range of
errors at low power regardless of the requested MAPE. We formalize these objectives as follows in Definition 1.1.

**Definition 1.1** A scenario set is said to be plausible if:

1. the shape of the error distribution for the scenarios is close to the shape of the empirical distribution of errors, i.e., its distribution plausibility score is close to 1 (as defined in later in Sect. 3.4);
2. the computed autocorrelation coefficients for the set are close the empirical values; and
3. the computed curvature for the set is close to the empirical value, especially when the scenarios are forecasts (because we observe that forecasts typically have lower curvature than actuals.)

We discuss ways to evaluate the plausibility of a scenario set and provide illustrations in Sect. 6.

## 2 Modeling the Joint Distribution of \((E, X)\)

The symbol \(E\) represents the random variable of the error between the input \(x\) and the corresponding dependent input \(y\). In this section, we model the fact that the error depends on the input \(x\) and does not have the property of ‘white noise’. Therefore, we must account for the fact that \(x\) is also a random variable that we will note \(X\). To model their joint distribution, we define the random variable \(Z = (E, X)\).

Here, \(Z\) denotes a random variable with values in \((-\infty, +\infty) \times (0, +\infty)\)—or, if the production capacity \(cap\) is known by the forecaster, values in \([-cap, cap] \times [0, cap]\). We denote by \(f_Z\) the density of \(Z\), and denote by \(f_E\) and \(f_X\) the marginals of \(f_Z\). Then,

\[
f_E(\varepsilon) = \int_{-\infty}^{\infty} f_Z(\varepsilon, x) dx, \quad f_X(x) = \int_{-\infty}^{\infty} f_Z(\varepsilon, x) d\varepsilon
\]

We also define the conditional density of \(E\) given \(X = x\) as:

\[
f_{E|X=x}(\varepsilon) = \frac{f(\varepsilon, x)}{f_X(x)}
\]

Modeling the conditional distribution of errors is important as it can vary significantly with the value of input data. For example, when the forecasts and the actuals are both low, the errors will be biased because the power produced cannot be below zero (in settings where power consumed by wind farms is both forecast and reported, the lower bound could be negative, but we use zero without loss of generality). Symmetrically, close to the maximum capacity, \(cap\), errors are bounded by the fact that power cannot exceed maximum production capacity.

In this context, we introduce the functional \(m(x)\) to denote the expected value of the absolute error of the distribution conditioned on \(x\), defined as:
We then introduce $r$ to denote the mean absolute relative error, defined as:

$$r = \mathbb{E}\left[\mathbb{E}\left[|\mathcal{E}| \mid X = x\right]\right] = \mathbb{E}\left[\frac{m(X)}{X}\right]$$

In Fig. 2, we provide an illustrative visualization of the distribution of the relative error $RE$ with respect to the actuals. We note that because the actuals are correlated with the forecasts, the figure would be very similar if the forecasts were used in instead. The data is for CAISO wind power data, ranging from July 1, 2013 to June 30, 2015. We will use this dataset for illustration throughout the paper, and refer to it informally as the CAISO Wind data set. These data are available in the mape_maker software distribution; the file is wind_total_forecast_actual_070113_063015.csv.

2.1 Estimating the Conditional Distribution of $\mathcal{E}|X, \hat{\mathcal{E}}|X=x$

In [2], it was shown that an appropriate probability density function for the wind power forecast error is the beta distribution. Besides, its variable kurtosis and its

![Fig. 2 Empirical joint distribution of $(\hat{\mathcal{E}}, X)$ - CAISO wind power](image_url)
bounds make it more suitable than the Normal distribution to our end. Finally, its simplicity (only two shape parameters) is useful for the estimation.

Given the notation \( x \in X \), we use the beta distribution on \([l, s + l]\) to model \( f_{\xi|X=x} \). In addition to the \( l \) and \( s \) that we will refer to as location parameters, a beta distribution requires two additional parameters—\( \alpha \) and \( \beta \), i.e., the shape parameters. We then define

\[
f_{\xi|X=x}(\varepsilon) = \text{beta}(\varepsilon; (\alpha, \beta, l, s)) = \frac{(\frac{\varepsilon - l}{s})^{\alpha-1}(1 - \frac{\varepsilon - l}{s})^{\beta-1}}{B(\alpha, \beta)}
\]

with

\[
B(\alpha, \beta) = \int_{\varepsilon=l}^{l+s} (\frac{\varepsilon - l}{s})^{\alpha-1}(1 - \frac{\varepsilon - l}{s})^{\beta-1}d\varepsilon
\]

### 2.2 Intervals for Conditional Estimation

We estimate the parameters of the conditional density for each \( x \) of the input dataset, \( X \), using a a fraction \( a \) (e.g., 0.05; see Sect. 2.4) of data before and after each \( x \). Let \( G_X \) denote the empirical cumulative distribution function.

Then, let \( I_x^a = [G_X^{-1}(G_X(x) - a), G_X^{-1}(G_X(x) + a)] \). Thus \( I_x^a \) is centered on \( \bar{x}(x; a) = \frac{G_X^{-1}(G_X(x) - a) + G_X^{-1}(G_X(x) + a)}{2} \) with 2\( a \) fraction of the data. We fit the shape parameters of a beta distribution on the sample \( E_{I_x} = \{\varepsilon_i, 1 \leq i \leq n, x_i \in I_x^a, \} \). Note that for production values near zero and near the capacity, there could be as few as \( a \) fraction of the values used.

To compute an estimate for a particular value \( x' \), our method uses the interval \( I_x^a \) for which \( \bar{x}(x; a) \) is closest to \( x' \) and finds the corresponding set \( E_{I_x^a} \) to compute the parameters for \( x' \). For every \( x' \in X \) that is not close to zero or \( cap \), the closest \( \bar{x}(x; a) \) to \( x' \) will often be just \( \bar{x}(x'; a) \). However, for very small or large values of \( x' \) and when \( X_{SID} \notin X \), the use of the interval with the closest mean is most appropriate.

We will now describe how our method fits the parameters of the beta distributions. Because every estimated quantity will depend on \( a \), we drop \( a \) as a subscript or function parameter for notational simplicity.

### 2.3 Fixing \( l, s \) and Estimating \( \alpha, \beta \)

#### 2.3.1 Constraints on the Location Parameters

An informed choice of the location parameters will avoid simulating errors leading to \( y \) values lower than 0 or greater then the \( cap \) of the dataset. We now define the function \( y_{max} \) which returns the maximum possible simulated value at \( x \) according to a conditional distribution \( f_{\xi|X=x} \). Because the inverse of the corresponding cumulative distribution function (CDF) evaluated at one, \( F_{\xi|X=x}^{-1}(1) = l + s \), is the maximum of the error
simulated; \( F_{\mathcal{X}|x=0}^{-1}(0) = l \) is the minimum; and because we want to avoid simulating values above the cap or below zero we have

\[
y_{\text{max}}(x) = x + F_{\mathcal{X}|x=0}^{-1}(1) \\
= x + s + l \\
\leq \text{cap}.
\]

Similarly,

\[
y_{\text{min}}(x) = x + F_{\mathcal{X}|x=0}^{-1}(0) \\
= x + l \\
\geq 0.
\]

These two conditions give

\[
\hat{l}(x) = \begin{cases} 
-x & \text{if } \min (\epsilon_i, x_i \in I_x) \leq -x \\
\min (\epsilon_i, x_i \in I_x) & \text{else}
\end{cases}
\]

\[
\hat{s}(x) = \begin{cases} 
\text{cap} - x - \hat{l}(x) & \text{if } \max (\epsilon_i, x_i \in I_x) \geq \text{cap} - x \\
\max (\epsilon_i, x_i \in I_x) - \hat{l}(x) & \text{else}
\end{cases}
\]

2.3.2 Choosing the Shape Parameters by the Method of Moments

The mean and variance of a beta(\(\alpha, \beta, l, s\)) distribution are:

\[
\mu = \frac{sl}{\beta + \alpha} + l \\
V = \frac{1}{s^2 (\alpha + \beta)^2 (\alpha + \beta + 1)} \frac{\alpha \beta}{(\alpha + \beta + 1)}
\]

We can now choose shape parameters by solving these two equations for \(\alpha\) and \(\beta\)

\[
\hat{\mu}(x) = \frac{\hat{s}(x) \alpha}{\beta + \alpha} + \hat{l}(x) \\
\hat{V}(x) = \frac{1}{\hat{s}(x)^2 (\alpha + \beta + 1)} \frac{\alpha \beta}{(\alpha + \beta)^2 (\alpha + \beta + 1)}
\]

to obtain \(\hat{\alpha}(x)\) and \(\hat{\beta}(x)\).

For any \(x \in \mathcal{X} \cup \mathcal{X}_{\text{SID}}\) assign

\[
\hat{S}_x = (\hat{\alpha}(x), \hat{\beta}(x), \hat{l}(x), \hat{s}(x))
\]
2.4 Selecting $a$

We now develop an empirical way to select the best $a$. If $a$ is small, the sample on which to fit the distribution will be small since $I_a^x$ is small. Fitting a distribution on very little data is of course dangerous. On the other hand, if $a$ is large, then the sample is too large to provide us with an estimation of the conditional density. In the extreme, where $a = 1$, every conditional density will be equal to the density of the relative error.

We select $a$ using a least squares fit based on a discrepancy score between the empirical distribution function and the one obtained by estimating each conditional distributions with $2a$ of the data. Let $g$ be the empirical joint density of $(X, \varepsilon)$. Let $\hat{f}$ be the joint density of $(X, \varepsilon)$ taken as $\hat{f}_{a}(x, \varepsilon) = \hat{f}(x) \ast \hat{f}_{a}(\varepsilon)$. We choose $a$ to minimize the deviation between the real density and the simulated density:

$$D^2(a) = \int_x \int_{\varepsilon} (g(x, \varepsilon) - \hat{f}_{a}(x, \varepsilon))^2 d\varepsilon dx.$$ 

3 Adjusting Conditional Densities to Fit a Target MARE

In order to satisfy the plausibility of objectives, we need to adjust the conditional densities so as to retain their shape and the relative contribution to the error across values of $x$.

- While $\mathcal{E}$ denotes the random variable describing the error with properties that can be estimated from $\mathcal{X}$, $\tilde{\mathcal{E}}$ denotes the simulated error as a random variable defined by a distribution that we will develop with desired properties for the simulation over $\mathcal{X}_{SID}$.
- In the previous section, we defined the population density, $f_{\mathcal{E}|X=x}$, and the estimated density, $\hat{f}_{\mathcal{E}|X=x}$. In this section, we introduce a third conditional distribution: the simulation density $\tilde{f}_{\mathcal{E}|X=x}$.

The simulation density distribution of $\tilde{\mathcal{E}}|X$ will be chosen to verify the plausibility objectives while ensuring that the expected relative absolute error of the simulated random variable $\tilde{\mathcal{E}}$ matches the desired MARE $\tilde{r}$:

$$\mathbb{E}_{\mathcal{E}} \left[ \frac{1}{n_{SID}} \sum_{x \in \mathcal{X}_{SID}} \frac{|\tilde{\mathcal{E}}| |X = x|}{x} \right] = \tilde{r}.$$ 

3.1 Adjusting the Location Parameters to Fit a Target MAE

We want to adjust each conditional distribution such that the global distribution of $\tilde{\mathcal{E}}$ satisfies the target MARE and maintains the same shape parameters as the original distributions. Toward this objective, we compute the mean absolute error of a beta
distribution when $\alpha$ and $\beta$ are fixed. First, let $l < 0$ and $s + l > 0$. Then, let $b(\cdot; \alpha, \beta, l, s)$ be an arbitrary beta density function with parameters $(\alpha, \beta, l, s)$ for which we define a mean absolute error function of $l$ and $s$ given values for $\alpha$ and $\beta$ as

$$v(l, s; \alpha, \beta) = \int_{\varepsilon = l}^{s + l} |\varepsilon| b(\varepsilon; \alpha, \beta, l, s) d\varepsilon.$$  

We then observe:

$$\lim_{s \to 0} v(l, s; \alpha, \beta) = 0, \quad \forall l < 0$$

$$v(l, s; \alpha, \beta) \sim \frac{s\alpha}{\pi \alpha + \beta}$$

Since $v$ is continuous (it is a sum of continuous functions), the intermediate value theorem applies which means that $v(l, s; \alpha, \beta)$ can achieve any value and in particular, the value needed in order to hit the specified error target.

Thus, once we are given $\alpha$, $\beta$, and a target value for the absolute error at a particular value of $x$, we need to find the intersection between a hyperplane defined by the target and the surface defined by $v(l, s; \alpha, \beta)$ to establish values for $\hat{l}$ and $\hat{s}$. For $x \in \mathbb{R}_+$ we will want to choose the solution that minimizes the distance to the estimated values $\hat{l}(x)$ and $\hat{s}(x)$ while hitting a target mean absolute error $m(x)$ and without changing the shape parameters.

This could be accomplished using

$$\begin{align*}
(\hat{l}(x), \hat{s}(x)) &= \arg\min_{l,s} (l - \hat{l}(x))^2 + (s - \hat{s}(x))^2 \\
& \text{s.t.} \quad l \in \mathbb{R}, \ s \in \mathbb{R}_+ \\
& \quad 0 \geq l \geq -x \\
& \quad 0 \leq s \leq \text{cap} - x - l \\
& \quad v(l, s; \hat{\alpha}(x), \hat{\beta}(x)) = m(x)
\end{align*}$$

but requiring no deviation from $m(x)$ can cause computational issues. Thus, for expediency, we instead use in our experiments the following formulation that yields very similar results:

$$\begin{align*}
(\tilde{l}(x), \tilde{s}(x)) &= \arg\min_{l,s} (v(l, s; \hat{\alpha}(x), \hat{\beta}(x)) - m(x))^2 + \left(\frac{l - \tilde{l}(x)}{o(x)}\right)^2 + \left(\frac{s - \tilde{s}(x)}{o(x)}\right)^2 \\
& \text{s.t.} \quad 0 \geq l \geq -x \\
& \quad 0 \leq s \leq \text{cap} - x.
\end{align*}$$

In this formulation, we employ $o(x)$ as a heuristic to scale the deviation for $l$ and $s$ to be less important than the deviation for $m$ using, for example, $o(x) = \max(|\tilde{l}(x)|, |\tilde{s}(x)|, m(x))$.

Since there are bound constraints on $l$ and $s$ (see Sect. 2.3.1), $v$ cannot hit every target $m(x)$. We compute a maximum target function that can be hit as:

$$m_{\text{max}}(x) = \max_{l \in (-x, 0], s \in [0, \text{cap} - x)} v(l, s; \hat{\alpha}(x), \hat{\beta}(x)).$$
The target function $m$ must then be bounded for every $x$ by:

$$m(x) \leq m_{\max}(x). \quad (3)$$

Given a mean absolute error target function $m$ satisfying inequality (3) we obtain for any $x$, a beta distribution of parameters $\tilde{S}_{x,m} = (\tilde{a}(x), \tilde{b}(x), \tilde{l}(x), \tilde{s}(x))$ that satisfies the mean absolute error target and that is the closest possible to the estimated distribution. We now proceed to allocate an error target to each $x \in \mathcal{X}_{SID}$ that we will call $\tilde{m}$ and that depends on the target MARE and on a weight function.

### 3.2 Changing the Conditional Distributions

#### 3.2.1 Weight Functions

Let’s define $\Omega_{\mathcal{X}_{SID}}$ as the set of functions $\omega_{\mathcal{X}_{SID}}$ defined on $\mathcal{X}_{SID}$ such that

$$\frac{1}{n_{SID}} \sum_{x \in \mathcal{X}_{SID}} \omega_{\mathcal{X}_{SID}}(x) = 1.$$  

We call them weight functions. Weight functions will be used to assign a target MAE to obtain from each of the conditional distributions $\tilde{E}(X = x)$, for all $x \in \mathcal{X}_{SID}$. It can also be seen as the function that weights the contribution of the Absolute Error of each conditional distribution to the Mean Absolute Relative Error of the simulation.

#### 3.2.2 Target Function Generator

We also define the following functional that we call target function generator.

$$\tilde{m} : \mathcal{X}_{SID} \times \mathbb{R}_+ \times \Omega_{\mathcal{X}_{SID}} \rightarrow \mathbb{R}_+ \quad \text{(Target function generator)}$$

For a fixed $\tilde{r}$ and $\omega$, $\tilde{m}(., \tilde{r}, \omega)$ is a target function for $\tilde{r}$ over the SID. Since the target function will be used to directly adjust the conditional distribution, it must respect the inequality (3). Finally, we say that a target MARE $\tilde{r}$ is feasible for a given $\omega \in \Omega_{\mathcal{X}_{SID}}$ if

$$\forall x \in \mathcal{X}, \tilde{m}(x, \tilde{r}, \omega) \leq m_{\max}(x)$$

#### 3.2.3 Zero Power Input

We recall that the zero input does not count in the computation of the MARE. However, we want the distribution of the simulated errors to be drawn from the estimated distribution. In other words :

$$\forall \tilde{r} \in \mathbb{R}_+, \quad \tilde{l}(0) = \hat{l}(0) \text{ and } \tilde{s}(0) = \hat{s}(0)$$

We assign
To avoid big discontinuities in the parameters of the beta distributions, we could take as 
\[ \tilde{l}(0) = \lim_{x \to 0} \tilde{l}(x), \tilde{s}(0) = \lim_{x \to 0} \tilde{s}(x). \]

### 3.2.4 Convergence to the Requested MARE

Using the function \( \tilde{m} \) to assign a target MAE for each SID input will allow us to hit the target MARE using the simulation distributions. Indeed, let us define the random variable \( \tilde{E}|X \) with density 
\[ \tilde{f}_{E|X}(\varepsilon) = \frac{b(\varepsilon, \tilde{S}_{x,\tilde{m}})}{\tilde{E} | X = x}, \varepsilon \in (-\text{cap}, \text{cap}). \]

If we establish the distribution parameters as described in Sect. 3.1 and solve Program (1) with 
\[ m(x) = \tilde{m}(x; \tilde{r}, \omega_{X_{\text{SID}}}) \] we have,

\[ \int_{-\infty}^{\infty} |\varepsilon| b(\varepsilon, \tilde{S}_{x,\tilde{m}}) d\varepsilon = \tilde{m}(x; \tilde{r}, \omega_{X_{\text{SID}}}), \forall x \in X_{\text{SID}}. \]

Then, the expected MARE with the errors drawn from these distributions and with the inputs in the \( X_{\text{SID}} \) is:

\[
\mathbb{E}_{\tilde{E}} \left[ \frac{1}{n_{\text{SID}}} \sum_{x \in X_{\text{SID}}} \frac{|\tilde{E}| |X = x|}{x} \right] = \frac{1}{n_{\text{SID}}} \sum_{x \in X_{\text{SID}}} \mathbb{E}_{\tilde{E}|X=X} \left[ \frac{|\tilde{E}| |X = x|}{x} \right] \\
= \frac{1}{n_{\text{SID}}} \sum_{x \in X_{\text{SID}}} \frac{\tilde{m}(x; \tilde{r}, \omega_{X_{\text{SID}}})}{x} \\
= \frac{\tilde{r}}{n_{\text{SID}}} \sum_{x \in X_{\text{SID}}} \omega_{X_{\text{SID}}}(x) \\
= \tilde{r} \\
\]

This is true with any weight function for which \( \frac{1}{n_{\text{SID}}} \sum_{x \in X_{\text{SID}}} \omega_{X_{\text{SID}}}(x) = 1 \). We now proceed to describe the construction of a sensible weight function.

### 3.3 Weight Function for \( X_{\text{SID}} = X \)

We recall the plausibility objectives: we want the distribution of the simulated errors to be as close as possible to the population distribution. In particular, suppose that we want to do a simulation with a target MARE that happens to be the same as the MARE of the original data and further suppose that we want to simulate using values from the entire data set (i.e., \( X_{\text{SID}} = X \)). Then we expect the simulated conditional distributions to be equal to the estimated conditional distributions. In other words,

\[ X_{\text{SID}} = X, \quad \tilde{r} = r_{\tilde{m}} \implies \forall x \in X, \tilde{l}(x) = \tilde{l}(x) \text{ and } \tilde{s}(x) = \tilde{s}(x). \]

Solving Program (2) defined in Sect. 3.1, leads to \( \tilde{l}(x) \approx \hat{l}(x) \) and \( \tilde{s}(x) \approx \hat{s}(x), \forall x \in X \).

Let us define the following \( \hat{\omega}_{\Lambda} \) function,

[ Springer]
By definition, we have \( \frac{1}{n} \sum_{x \in X} \tilde{w}_X(x) = 1 \). It is a weight function.

The choice of this weight function is natural when \( X_{SID} = X \) because it is the ratio of the expected absolute relative error simulated at \( x \) over the mean absolute relative error when the errors are distributed according to the estimated joint distribution. However, choosing it when \( X_{SID} \neq X \) would satisfy our requirement for plausibility but it would prevent us from hitting the requested MARE.

Figure 3 illustrates that for the full CAISO wind dataset, the weight function presents a hyperbolic shape. The low values account for the biggest part of the MAPE.

### 3.4 Weight Function for \( X_{SID} \) and an Arbitrary \( \tilde{r} \)

Let us define the following real number that we call the distribution plausibility score:

\[
P_{X_{SID}} = \frac{1}{n_{SID}} \sum_{x \in X_{SID}} \tilde{w}_X(x)
\]

When \( X_{SID} \neq X \), the distribution of the SID is different from the distribution of the input dataset. Thus we do not necessarily have \( P_{X_{SID}} = 1 \). A goal of our method is to meet the requested MARE, at least in expectation, no matter the \( X_{SID} \). If \( P_{X_{SID}} \) is greater than 1, it means that the distribution of \( X_{SID} \) has more data in the range where the weight function takes high values. This means that if use \( \tilde{w}_X \), we are going to simulate too many errors with high values. While it makes some physical sense, we are nonetheless going to simulate a greater MARE than expected. Symmetrically,
if \( P_{X_{SID}} \) is smaller than one, we are going to retrieve a lower MAPE than expected. This is illustrated in Fig. 4 where the density for the \( X_{SID} \) between December 2013 and March 2014 indicates more values at lower power than for the entire dataset, \( X \). If we simply used \( \hat{\omega}(x) \), for \( x \in X_{SID} \), then meeting the target AREs for each \( x \) would result in a MARE much greater than specified. In other words, since the ARE/MARE ratio is very high for the low power input, and since these inputs are over represented under the distribution of the December 2013–March 2014 SID, we are going to simulate too many errors with a high target of mean absolute error. To meet the target MARE, a re-scaled weight function must therefore be computed.

Let us define the following SID weight function:

\[
\forall x \in X_{SID}, \quad \tilde{\omega}_{X_{SID}}(x) := \frac{\hat{\omega}(x)}{P_{X_{SID}}}
\]

With the re-scaled factor, we have \( \frac{1}{n_{SID}} \sum_{x \in X_{SID}} \hat{\omega}_{X_{SID}}(x) = 1 \) so \( \tilde{\omega}_{X_{SID}} \in \Omega_{X_{SID}} \).

Finally, for a given feasible \( \tilde{\tau} \in \mathbb{R}_+^* \), we compute a \( \tilde{\omega}_{X_{SID}} \) which allocates the absolute errors across \( X_{SID} \) based on the allocation across \( X \). With these two parameters we can compute \( \tilde{m}(x; \tilde{\tau}, \hat{\omega}_{X_{SID}}), x \in X_{SID} \). According to Sect. 3.2.4, defining \( \mathcal{E} \) from this target function, will get us \( \mathbb{E}[\mathcal{E}] = \tilde{\tau} \).

We can also get the feasibility region for the target MARE. For a given \( \tilde{\tau} \) to be a feasible target MARE, it must satisfy \( \forall x \in X, \quad \tilde{m}(x, \tilde{\tau}, \hat{\omega}_{X_{SID}}) \leq m_{max}(x) \). We can deduce a feasibility region for the target MARE:

\[
\tilde{\tau} \leq \tilde{R}_{X_{SID}} = P_{X_{SID}} \min \left( \frac{m_{max}(s)}{s \hat{\omega}(s)}, \ s \in X_{SID} \right)
\]

![Fig. 4](image.png) Comparison test density versus all dataset density
3.5 Simulating Without autocorrelation

Simulation of forecast errors without autocorrelation is now straightforward. We generate a vector that is identically and independently distributed (i.i.d.) following a Uniform on \([0, 1]\), \(\tilde{\varepsilon}_t, i \leq n_{SID}\). Then,

\[
\tilde{\varepsilon}_{t,i} = \hat{F}^{-1}_{X=X_{X,X}(\tilde{u}_{t,i})}, \forall i \leq n_{SID}
\]

and

\[
\tilde{y}_{t,i} = x_{t,i} + \tilde{\varepsilon}_{t,i}, \forall i \leq n_{SID},
\]

so

\[
\mathbb{E}[MARE(x_{SID}^t, \tilde{y}_t)] = \tilde{r}.
\]

While we are hitting the target MARE in expectation, the entire autocorrelation of the simulated errors relies solely on the autocorrelation of the input. In the case where the errors are not depending on the input, i.e. \(\hat{F}_{X=X^X} = \hat{F}_{X=X}^X, \forall i \leq n_{SID}\) - which is the case for the middle power range for the CAISO wind data—then our simulations would have a null autocorrelation function (except for lag 0). It would resemble a white noise process, which is not plausible according to our criteria. Therefore, we need to implement a base process that will replace \((\tilde{u}_{t,i})_{i \leq n_{SID}}\) and will generate the needed autocorrelation to satisfy the second point of the plausibility criteria.

4 Inferring a Base Process

Simulating forecast errors that have the estimated beta distribution as well as the appropriate temporal autocorrelation requires the estimation and use of a base process. Instead of simply passing i.i.d. uniform pseudo-random numbers to the inverse of the beta distributions, we create a base process in a way that is similar to the methods used in [13]. Our method uses a base process to generate a pseudo-random vector \(\hat{u}_t = (\hat{u}_{t,i})_{i \leq n_{SID}}\) that is marginally Uniform on \([0,1]\) and that depends on the past p lags of the base process and the past q lags of errors over the base process. Then, as seen previously in Sect. 3.5, we simulate the forecast errors via the transformation \(\hat{F}^{-1}_{X=X} (\hat{u}_t)\).

In order to obtain a distribution that enables creation of a vector of Uniform pseudo random numbers with the appropriate characteristics, we use techniques inspired by the ARTA fit method (see [1]). The method makes use of the CDF for the standard normal distribution, which we denote using \(\phi\). The CDF of the conditional distribution \(\tilde{\varepsilon}\mid X\), which is a beta distribution fit using \(X\) is denoted by \(\hat{F}_{\tilde{\varepsilon}X}\). Let us define the following time series \(\tilde{z}_t\), as the base process time series of the dataset:
\[ \forall i \leq n, \quad \hat{z}_{i,i} = \phi^{-1}(\hat{F}_{E|X=x_i}(\epsilon_{i,i})) \]

Its empirical distribution is close to a standard Gaussian. Indeed, in Sect. 2.3 we are estimating the conditional distribution so that \( \mathcal{E}|X = x \) has distribution that is approximated by \( \hat{F}_{E|X=x} \), thus, \( \hat{F}_{E|X=x}(\mathcal{E}) \sim \mathcal{U}(0, 1) \) and \( \hat{z}_i \sim N(0, 1) \). We fit on this base process an ARMA process. The standard definition of an ARMA process of order \( p \) and \( q \) uses \( (a_i)_{i\leq p} \) and \( (b_i)_{i\leq q} \) as coefficients so we temporarily reuse those symbols in this section.

**Definition 4.1** \( \{z_i\} \) is a base process if

- \( \{z_i\} \) follows an ARMA process of order \( p \) and \( q \):
  \[
  z_i = \sum_{h=1}^{p} a_h z_{i-h} + \sum_{h=1}^{q} b_h \delta_{i-h} + \delta_i
  \]
  Where \( \{\delta_i\} \) are i.i.d. Gaussian random variables with mean 0 and variance \( \sigma^2_{\delta} \).
- \( \text{Var}[z_i] = 1, E[z_i] = 0 \), so that for all \( i, z_i \sim N(0, 1) \).

We run a grid search over multiple \( (p, q) \) and we select the ARMA model to minimize the BIC criterion. The \( (a_i)_{i\leq p} \) and \( (b_i)_{i\leq q} \) found during the process define a function that enables us to generate base processes which will create the autocorrelation that we are looking for. We further specify \( \sigma_{\delta} \) so that we get \( \text{Var}[\hat{z}_i] = 1 \).

Then, we can simulate the error directly by
\[
\hat{e}_{i,i} = \hat{F}_{E|X=x_i}^{-1}(\phi(\hat{F}_{E|X=x_i}(\epsilon_{i,i}))), \quad \forall i \leq n_{SID}
\]

and also get the result for the expected MARE established in Sect. 3.5.

By using an ARMA process as a base process we get the desired autocorrelation without the assumption of Gaussian errors that accompanies direct use an ARMA process [1]. As an aside, we note that the textbook use of i.i.d. uniforms as described in Sect. 3.5 can be seen as a degenerate case where the base process is of order \( p = 0 \) and \( q = 0 \).

## 5 Enforcing Curvature

Let \( (y_i) \in \mathbb{R}^n \) denote a simulation output time series. We define the **curvature** at a point \( i \) in \( (y_i) \) as
\[
s_i = y_{i+2} - 2y_{i+1} + y_i, \quad \forall i < n - 2,
\]
i.e., a second difference.

Methods described in Sect. 4 successfully model temporal correlation between the errors while satisfying a target MARE. However, some scenarios might not “look right” because of their lack of smooth curvature. This is specified as the third point of the plausibility criteria. This is especially unsatisfying in the case

\[ \text{Springer} \]
of forecasted renewables power production, which are much less sharp and erratic when compared to actual quantities.

We now concretely illustrate this issue in Fig. 5 with analysis of the CAISO wind power production data introduced previously. In Fig. 5a, we show baseline scenarios resulting from our proposed methods, i.e., without adjustment for curvature. As is clearly observed, the simulated forecasts in this case closely mirror the actuals—and not the one “true” forecast. In contrast, we show in Fig. 5b a closely related set simulated forecasts—obtained by the procedure we now describe—that instead exhibit significantly smoother and more realistic-looking curvature. Ultimately, the need for such adjustment depends entirely on the application.

In order to adjust the curvature of a forecast while still achieving a target MARE, one approach is to a posteriori adjust a time series that already satisfies a target MARE such that specific curvature characteristics are imposed. We now formalize this general approach.

We introduce a minimization problem in which we penalize deviations from both a target second difference and the simulated forecast error. Per earlier analysis, we can simulate \( \hat{\epsilon}_i \) using an ARMA base process. Then, define \( d \in \mathbb{R}^+ \) as the target second difference, and we let the user decides on the penalization weights: \( W_s \) and \( W_\epsilon \) in \( \mathbb{R}^+ \).

We then let \( (y_i) \) denote the solution of the following mathematical program:

\[
\begin{align*}
\min_y & \quad \sum_{i=3}^{n} W_s \left( |y_i - 2y_{i-1} + y_{i-2}| - d \right)^2 + W_\epsilon \left( y_i - x_i - \hat{\epsilon}_i \right)^2 \\
\text{s.t.} & \quad y \in [0, \text{cap}]^n
\end{align*}
\]

For practical computation, we now transform this mathematical program so that the objective function is quadratic and constraints are linear—such that widely available mathematical programming solvers can be leveraged. The transformation yields the following equivalent mixed-integer linear program (MILP), with \( 3n \) additional variables, \( n \) equality constraints, and \( 3n \) inequality constraints (\( 6n \) if we consider that the three real vectors are negatively bounded by 0):

\[
\begin{align*}
\min_{y, \lambda^+, \lambda^-, b} & \quad \sum_{i=2}^{n} W_s \left( \lambda^+_i + \lambda^-_i - d \right)^2 + W_\epsilon \left( y_i - x_i - \epsilon_i \right)^2 \\
\text{s.t.} & \quad y \in \mathbb{R}_+^n, \quad \lambda^+ \in \mathbb{R}_+^n, \quad \lambda^- \in \mathbb{R}_+^n, \quad b \in \{0, 1\}^n \\
& \quad y_i \leq \text{cap} \\
& \quad \lambda^+_i - \lambda^-_i = y_i - 2y_{i-1} + y_{i-2}, \quad \forall i \leq n \\
& \quad \lambda^+_i \leq b_id_{\text{max}} \\
& \quad \lambda^-_i \leq (1 - b_i)d_{\text{max}}
\end{align*}
\]

where \( d_{\text{max}} \) denotes a large constant; a safe value is four times the capacity.

To verify equivalence of the two mathematical programs, we note that if \( y_i - 2y_{i-1} + y_{i-2} \geq 0 \), then because \( b_i \in \{0, 1\} \), \( \lambda^-_i \) is equal to 0 with the two last equations. Then \( \lambda^+_i = y_i - 2y_{i-1} + y_{i-2} \) and \( \lambda^+_i + \lambda^-_i = y_i - 2y_{i-1} + y_{i-2} \). We use the same reasoning when \( y_i - 2y_{i-1} + y_{i-2} < 0 \).

Numerous open source and commercial solvers are available for such a mathematical program. However, solution time does generally increase with \( n \). In many
applications the restrictions on curvature are motivated by aesthetic or heuristic considerations. Thus, it can be reasonable to specify a “loose” optimality gap to avoid excessive computation time.

6 Evaluation

In this article, we proposed three types of simulations all based on the conditional estimation and adjustment of Sect. (3.2). We then added that we could safely model a base process to be one step closer to the autocorrelation of the observed dependent variable, and that—eventually—by correcting the curvature we could have smooth forecast simulations. We denote those three procedures:

- (A) "IID" as the base process which is i.i.d.,
• (B) "ARMA" as the base process which follows an ARMA process,
• (C) "ARMA + curvature" as the base process which follows an ARMA process and we correct the curvature \textit{a posteriori},

6.1 Behavior of the Simulated MAREs as $M$ Grows

To illustrate the behavior of the simulated scenarios as the number of scenarios created, $M$, grows we conducted experiments using the CAISO wind dataset and created scenarios for 3 days. Figure 6 shows that for this example, the achieved MARE is close to the target MARE as soon as there are about 5 scenarios.

6.2 Behavior of the Curvature and Autocorrelation Function of the Simulations

Let $p$ be the maximum lag of autocorrelation we wish to assess.

Let us define the value of the autocorrelation function of $\epsilon_i$ at lag $j$,

$$\hat{\rho}(\epsilon_i, j) = \frac{1}{(n-j)\sigma^2} \sum_{i=0}^{n-j} \epsilon_{t,i+j}\epsilon_{t,i}$$

From there, we can define a score, the squared root of the sum of the squared deviation of the autocorrelation of the simulation with the observed errors:

$$S_{\text{auto\_correlation}}(\epsilon_i, \tilde{\epsilon}_i) = \sqrt{\sum_{j=1}^{p} (\hat{\rho}(\epsilon_i, j) - \hat{\rho}(\tilde{\epsilon}_i, j))^2}$$

![Fig. 6 MARE score as a function of the number of scenarios created by simulation](image-url)
We ran $M = 20$ simulations over the CAISO wind dataset and created scenarios for a target MAPE of 200% and for data ranging from 2013-07-01 to 2015-06-30 (Table 1).

First we confirm that each procedure results in a close MARE with respect to the targeted MARE. The scenarios that use curvature correction result in a value that is closest to the target, which makes sense because Program 5 corrects for the MARE after the scenarios are created. However, the other scenarios are reasonably close. We also notice that for base processes that are not i.i.d., the autocorrelation score is quite good almost regardless of the number of scenarios. Finally, we can notice that for each value (MARE, autocorrelation score and Curvature) the methods are ordered as expected.

### 7 Conclusions and Future Directions

We have described methods for creating scenarios that make use of a history of forecast errors. The methods are summarized in algorithm form in the Appendix and the corresponding software is available for download and use. Although we used wind data from CAISO in our illustrations, the method can be used for any situation where there is a history of forecasts and actuals. In particular, the software has been used to create scenarios for load, solar, and wind for the rts-gmlc data https://github.com/GridMod/RTS-GMLC.

The use of solar requires pre- and post-processing of the input data to work well. Instead of power values, the forecasts and actuals should be presented as fractions of capacity and with the value of \textit{cap} set to one during the day and zero at night. This is because solar power is always zero at night and because the concept of “low power” changes during the day.

This work enables new research in areas such as:

1. Rigorous quantification of the potential benefits associated with improvements to renewables power production forecasts, to determine if the costs associated with procurement of improved numerical weather prediction (NWP) models and

### Table 1

The first two lines indicate the values observed in the input dataset and the target values for the scenarios

| MARE      | $S_{auto\_correlation}$ | Curvature |
|-----------|--------------------------|-----------|
| Observed  | 3.77162                  | 0         | 71.658 |
| Targeted  | 2                        | 0         | 71.658 |
| IID       | 1.896 ± 0.142            | 1.949 ± 0.025 | 370.306 ± 2.091 |
| ARMA      | 1.914 ± 0.215            | 0.011 ± 0.004 | 169.071 ± 1.332 |
| ARMA + curvature | 1.888 ± 0.298 | 0.011 ± 0.022 | 85.531 ± 19.625 |

The rest of the table gives the mean ± the standard deviation for 20 scenarios simulated with respect to three procedures.
associated hardware to provide boundary / initial conditions (e.g., meteorological towers) translate to improved system reliability and reduced operations costs.

2. Analyses of specific events in which forecast error was significantly higher than typically observed, e.g., during extreme weather conditions. A recent real-world motivating case stems from significant wind forecast errors observed by the Mid-continent Independent System Operator (MISO) in the US during polar vortices.

3. Bootstrapping to produce realistic alternative synthetic actuals for renewables power production, to mitigate issues associated with availability of limited (e.g., a year) historical data when conducting power system operations simulations; very large numbers of samples are required for rigorous reliability and cost-benefit analyses.

Future research includes consideration of error measures other than the MAPE. On the purely software front, we are working to parallelize computations. The software and the methods described here are intended to be an addition to the kit of tools available for dealing with uncertainty in power generation planning and operations.

Acknowledgements This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes. We are grateful for the referees’ comments, which improved the paper.

Putting It All Together

In this section we summarize the process to deliver a simulation with correct targets.

Procedures for Estimation

First, as shown in Algorithm 1, we preprocess the data and estimate the conditional distributions using the methods explained in Sect. 2.3. This results in a set of beta distribution parameters for each input from the whole dataset, called $\hat{S}$. To estimate the parameters we recall that the user should specify a data fraction for the sampling (e.g., 0.05). (The software provides an option to produce a curve for the scores described in Sect. 2.4.)
Next, as shown in Algorithm 2, we estimate the partitioning of the mean absolute percent errors according to the input and we encode this information in the weight function. An important feature of this procedure is the computation of $\tilde{r}_m$ which is the expected mean absolute relative error from the conditional distributions (which may be close in value to, but is different from, $\hat{r}$.) This procedure is explained in Sect. 3.3.

The next phase, shown in Algorithm 3, is estimation of the underlying base process that generates autocorrelation in the time series of the errors. This is done by using the CDF $B$ of the beta distribution whose parameters have been inferred in step 1. Then we operate a grid search over the $p$ and $q$ parameters to select the order of the model that minimizes the BIC criterion. We save the coefficients. Recalling that we want the marginal of $Z$ to follow a standard normal, we set the variance of the errors of the base process so that $\text{Var}(Z_i) = 1$. This procedure is explained in Sect. 4.
Algorithm 3 Fitting the Base Process ARMA process

Input: $x_t$, $\varepsilon_t$, $\hat{S}_x$
Output: $(a_i)_{i \leq p}$, $(b_i)_{i \leq q}$, $\sigma_s$

1: procedure $\text{Fit_{ARMA\_Process}}(x_t, \varepsilon_t, \hat{S}_x)$
2: for $i \in [1, \text{len}(x)]$ do $\triangleright$ Estimating the base process see Section 4
3: $\hat{\sigma}_{t,i} \leftarrow \phi^{-1}(B(\hat{\varepsilon}_{t,i}, \hat{S}_x(x_i)))$
4: $\text{BIC} \leftarrow +\infty$
5: $p$, $q$ $\leftarrow$ 0, 0
6: for $p', q' \in [0, 5]^2$ do $\triangleright$ Grid Searching
7: $\text{tempBIC} \leftarrow \text{BIC}(\text{ARMA}(\hat{\sigma}_{t}, (p', 0, q'))$
8: if $\text{tempBIC} < \text{BIC}$ then
9: $\text{BIC} \leftarrow \text{tempBIC}$
10: $p$, $q$ $\leftarrow$ $p'$, $q'$
11: $(a_i)_{i \leq p}$, $(b_i)_{i \leq q}$ $\leftarrow$ $\text{ARMA}(\hat{\sigma}_{t}, (p, 0, q))$
12: $\sigma_s \leftarrow \text{argmin}_x (\text{std}(\text{ARMA}((a_i)_{i \leq p}, (b_i)_{i \leq q}, \sigma) - 1)^2$
13: return $(a_i)_{i \leq p}$, $(b_i)_{i \leq q}$, $\sigma_s$

Procedures to Deliver the Target MARE

First, as shown in Algorithm 4, given a target MARE $\bar{r}$, and a $X_{SID}$ we verify that $\bar{r}$ is feasible. If it is, we aim at targeting a mean absolute error for each conditional distribution with input in the $X_{SID}$. For this, we compute a target function using the estimated weight function (see Sect. 3.4).

Algorithm 4 Inferring a target function for the SID

Input: $\hat{S}$, $X_{SID}$, $\bar{r}$, $\hat{\omega}_X$
Output: $\bar{m}$

1: procedure $\text{Computing\_Simulation\_Target\_Function}(\hat{S}, X_{SID}, \bar{r}, \hat{\omega}_X)$
2: $P_{X_{SID}} \leftarrow 0$ $\triangleright$ Computing the Distribution plausibility score
3: for $s \in X_{SID}$ do
4: $P_{X_{SID}} \leftarrow P_{X_{SID}} + \frac{\hat{\omega}_X(s)}{|X_{SID}|}$
5: $\bar{r}_{\text{max}} \leftarrow P_{X_{SID}} \min \left( \frac{\text{max}(s)}{\hat{\omega}_X(s), s \in X_{SID}} \right)$
6: if $\bar{r} > \bar{r}_{\text{max}}$ then
7: Report Error
8: $\hat{\omega}_{X_{SID}} \leftarrow \frac{\hat{\omega}_X}{P_{X_{SID}}}$
9: for $s \in X_{SID}$ do $\triangleright$ Applying the function as explained in Section 3.4
10: $\bar{m}(s) \leftarrow \frac{s \hat{\omega}_{X_{SID}}(s)}{\hat{\omega}_X(s)}$
11: return $\bar{m}$

Second, as shown in Algorithm 5, according to a target function $\bar{m}$, we assign adjusted parameters for each conditional distribution whose input is in the $X_{SID}$. We adjust the location parameters from the estimated values while keeping the shape parameters. See Sect. 3.2.
Using methods summarized in Algorithm 6, we simulate a base process sample of length $|\mathcal{X}_{SID}|$ and use the simulated conditional distributions to obtain conditioned errors. We directly get the simulation by summing the errors and the input data. Finally, if the user asks for it, we optimize the curvature a posteriori, see Sect. 5.

\begin{algorithm}
\caption{Inferring the simulation beta distributions}
\begin{algorithmic}[1]
\Require $\tilde{m}, \mathcal{S}, \mathcal{X}_{SID}$
\Ensure $\tilde{S}_n$
\Procedure{ADJUSTING\_SIMULATION\_PARAMETERS}{$\tilde{m}, \mathcal{S}, \mathcal{X}_{SID}$} \Comment Applying the methodology explained in section
\For{$\delta \in \mathcal{X}_{SID}$} \Comment not necessarily $\mathcal{X}_{SID} \subset \mathcal{X}$
\State $x \leftarrow \text{closest}(\delta, \mathcal{X})$
\State $\tilde{\alpha}(\delta), \tilde{\beta}(\delta) \leftarrow \alpha(x), \beta(x)$
\State $\tilde{l}(\delta), \tilde{s}(\delta) \leftarrow \text{Program1}(\tilde{\alpha}(\delta), \tilde{\beta}(\delta), \delta, \tilde{m}(\delta))$ \Comment See equation 2
\State $\tilde{S}_{n,\tilde{m}} = (\tilde{\alpha}(\delta), \tilde{\beta}(\delta), \tilde{l}(\delta), \tilde{s}(\delta))$
\EndFor
\State \textbf{return} $\tilde{S}_n$
\EndProcedure
\end{algorithmic}
\end{algorithm}

\begin{algorithm}
\caption{Simulating a sample of output}
\begin{algorithmic}[1]
\Require $\tilde{m}, \mathcal{S}, x_{i,\text{SID}}^{\text{SID}}, (a_i)_{i \leq \sigma}, (b_i)_{i \leq \sigma}, \sigma, \tilde{S}_{x,\text{SID}}$, which implies $\tilde{F}_x^{-1}$
\Ensure $(\tilde{y}_{t,i})_{i \leq \sigma, \text{SID}}$
\Procedure{COMPUTING\_ESTIMATION\_PARAMETERS}{$\tilde{m}, \mathcal{S}, \mathcal{X}_{SID}, (a_i)_{i \leq \sigma}, (b_i)_{i \leq \sigma}, \sigma$} \Comment $\mathcal{X}_{SID} \subset \mathcal{S}$
\For{$i \in [1, \sigma]$} \Comment $\tilde{F}_x$ \Comment $x_{i,\text{SID}}$
\State $\tilde{z}_{i,i} \leftarrow \text{createArmaSample}((a_i)_{i \leq \sigma}, (b_i)_{i \leq \sigma}, \sigma, \tilde{S}_{x,\text{SID}})$
\State $\tilde{r}_{i} = \tilde{F}_x^{-1}(x_{i,\text{SID}}^{\text{SID}}, \phi(\tilde{z}_{i,i}))$
\State $\tilde{y}_{t,i} = x_{t,i}^{\text{SID}} + \tilde{r}_{i,t}$
\EndFor \Comment Curvature is True then
\State $(\tilde{y}_{t,i})_{i \leq \sigma, \text{SID}} \leftarrow \text{Optimization1}(\tilde{z}_{i,i}, d, x_{i,\text{SID}}^{\text{SID}}, \text{cap})$ \Comment See Program 5
\State \textbf{return} $(\tilde{y}_{t,i})_{i \leq \sigma, \text{SID}}$
\EndProcedure
\end{algorithmic}
\end{algorithm}

\begin{thebibliography}{99}
\bibitem{Biller2005} Biller, Bahar, Nelson, Barry L.: Fitting time-series input processes for simulation. Oper. Res. \textbf{53}, 549–559 (2005)
\bibitem{Bludszuweit2008} Bludszuweit, Hans, Domínguez-Navarro, José Antonio, Llombart, Andrés: Statistical analysis of wind power forecast error. IEEE Trans. Power Syst. \textbf{23}(3), 983–991 (2008)
\bibitem{Dvorkin2014} Dvorkin, Y., Wang, Y., Pandzic, H., Kirschen, D.: Comparison of scenario reduction techniques for the stochastic unit commitment. In: 2014 IEEE PES General Meeting/Conference Exposition, pp. 1–5 (2014)
\bibitem{Feng2015} Feng, Yonghan, Rios, Ignacio, Ryan, Sarah M., Spurkel, Kai, Watson, Jean-Paul, Wets, Roger J.-B., Woodruff, David L.: Toward scalable stochastic unit commitment. Part 1: Load scenario generation. Energy Syst. \textbf{6}(3), 309–329 (2015)
\end{thebibliography}
5. Kolassa, S., Martin, R.: Percentage errors can ruin your day (and rolling the dice shows how). Foresight Int. J. Appl. Forecast. 23, 21–27 (2011)
6. Morales, Juan M., Minguez, Roberto, Conejo, Antonio J.: A methodology to generate statistically dependent wind speed scenarios. Appl. Energy 87(3), 843–855 (2010)
7. Pinson, P., Kariniotakis, G., Nielsen, H.A., Nielsen, T.S., Madsen, H.: Properties of quantile and interval forecasts of wind generation and their evaluation. In: Proceedings of the European Wind Energy Conference mnd Exhibition, Athens (2006). http://www.ewe.org
8. Pinson, P., Madsen, H., Nielsen, H.A., Papaefthymiou, G., Klöckl, B.: From probabilistic forecasts to statistical scenarios of short-term wind power production. Wind Energy 12, 51–62 (2008)
9. Pinson, Pierre, Girard, Robin: Evaluating the quality of scenarios of short-term wind power generation. Appl. Energy 96, 12–20 (2012)
10. Pinson, Pierre, Madsen, Henrik, Nielsen, Henrik Aa, Papaefthymiou, George, Klöckl, Bernd: From probabilistic forecasts to statistical scenarios of short-term wind power production. Wind Energy 12(1), 51–62 (2009)
11. Pinson, P., Tastu, J.: Discrimination ability of the energy score. Technical Report (2013)
12. Sari, Didem, Lee, Youngrok, Ryan, Sarah, Woodruff, David: Statistical metrics for assessing the quality of wind power scenarios for stochastic unit commitment. Wind Energy 19(5), 873–893 (2016)
13. Sharma, Kailash Chand, Jain, Prerna, Bhakat, Rohit: Wind power scenario generation and reduction in stochastic programming framework. Electric Power Compon. Syst. 41(3), 271–285 (2013)
14. Staid, Andrea, Watson, Jean-Paul, Wets, Roger J.-B., Woodruff, David L.: Generating short-term probabilistic wind power scenarios via nonparametric forecast error density estimators. Wind Energy 20(12), 1911–1925 (2017)
15. Wang, Jianhui, Shahidehpour, Mohammad, Li, Zuyi: Security-constrained unit commitment with volatile wind power generation. IEEE Trans. Power Syst. 23(3), 1319–1327 (2008)
16. Woodruff, David L., Deride, Julio, Staid, Andrea, Watson, Jean-Paul, Glevogt, Gerrit, Silva-Monroy, Cesar: Constructing probabilistic scenarios for wide-area solar power generation. Sol. Energy 160(15), 153–167 (2018)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.