Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft

Synthetic random environmental time series generation with similarity control, preserving original signal's statistical characteristics

Ofek Aloni [a], Gal Perelman [b], Barak Fishbain [b],*

[a] *Faculty of Mathematics, Technion - Israel Institute of Technology, Technion City, Haifa, 320003, Israel*
[b] *Faculty of Civil and Environmental Engineering, Technion - Israel Institute of Technology, Technion City, Haifa, 320003, Israel*

ARTICLE INFO

ABSTRACT

Synthetic datasets are widely used in applications like missing data imputation, simulations, training data-driven models, and system robustness analysis. Typically based on historical data, these datasets need to represent specific system behaviors while being diverse enough to challenge the system with a broad range of inputs. This paper introduces a method using discrete Fourier transform to generate synthetic time series with similar statistical moments to any given signal. The method allows control over the similarity level between the original and synthetic signals. Analytical proof shows that this method preserves the first two statistical moments and the autocorrelation function of the input signal. It is compared to ARMA, GAN, and CoSMoS methods using various environmental datasets with different temporal resolutions and domains, demonstrating its generality and flexibility. A Python library implementing this method is available as open-source software.

1. Introduction

The development of computational models to analyze complex environmental and infrastructure systems depends to a great extent on the availability of datasets with sufficient size and quality. In many cases, these kinds of datasets are not available, so that synthetic data generators that can accurately mimic the properties of the real data are crucially needed. This is true for a wide range of environmental problems such as studies of air quality (Moshenberg et al., 2015), wind speed (Negra et al., 2008), temporal and spatial analysis of fires (Hill et al., 2006), and weather characterization (Kilsby et al., 2007). Synthetic data are called for in cases of data unavailability, such as poorly recorded water demands time series, or too short historical datasets, that are not sufficient for computational methods (Nowak et al., 2010). Synthetic data are also commonly used for gap-filling, where only partial measurements are available (Kofinas et al., 2018). Another use of synthetic data is for examination of non-stationary scenarios which extend beyond the range of the historical records (Herman et al., 2016). Even when data is available and stationary, a synthetic data generator is very useful for the analysis and optimization of models, since it can be harnessed to evaluate system behavior and robustness in response to different and broad ranges of scenarios.

To give a more concrete example, robustness of water resource systems has recently attracted a great deal of attention. Although there are many definitions of robustness in the literature, most concur that it reflects system performance over a set of future scenarios (McPhail et al., 2021). Thus, in order to quantify robustness, a wide range of scenarios need to be examined. These scenarios should cover the range of values that have been observed in the past as well as future scenarios that include extreme values.

Generally speaking, synthetic data should comply with two requirements: the data should reflect the nature of the known parameters of historical records (time series) and at the same time allow for the manipulation of certain features to model non-stationary, less expected scenarios (Kang and Lansey, 2014). In physical systems, the first requirement is especially crucial, as the simulated system is bounded by the laws of physics. To illustrate, one can think of a water distribution system, where the standard deviation (STD) of water demands affects the hourly distribution of the demands. Synthetic scenarios with different STDs might not reflect a reasonable behavior such as peak flows appearing in the middle of the night. Another example is the creation of water hammers due to abrupt changes in the synthetic flows that might effect the system dramatically. In ambient simulations, drastic changes in humidity or temperature are simply not feasible.

Many studies have tackled the challenge of developing a generalized synthetic data generator. Early approaches included parametric methods such as Auto Regressive (AR) models and the Auto Regressive Moving Average (ARMA and ARIMA) models (Box and Jenkins, 1990), which are still common today. For example, Papamichail and Georgiou (2001) generated a synthetic monthly reservoir inflow series using the Seasonal ARIMA (SARIMA) model. Talbot et al. (2020) coupled ARMA with a Fourier series to generate electricity grid demands.

Non-parametric approaches for synthetic data generation are based on resampling and reshuffling historical data values. One of the most common non-parametric approaches is bootstrapping (Efron, 1992). This includes the Moving Block Bootstrap (MBB), the Circular Block Bootstrap (CBB), and the Stationary Bootstrap (SB). To take into account the original signal's autocorrelation, the bootstrap resampling is carried out in blocks of consecutively sampled values (Hirsch et al., 2015); (Selle and Hannah, 2010). However, block bootstrapping cannot handle the correlation between blocks. Another type of bootstrap method is the K Nearest Neighbor bootstrap (K-NN) which does maintain the block correlation (Nowak et al., 2010). However, all bootstrap methods are based on resampling the input data records. This ensures almost exactly the same statistical properties as the input, but causes the synthetic dataset to consist only of values found in the historical signals, and hence the generated time series has no new values.

A more recent approach for generating synthetic time series is based on data-driven models, such as Neural Networks (NNs). To this end, NN based approaches have been presented for water demand Odan and Reis (2012) and water quality forecasting (Chen et al., 2020). A recent data-driven approach for synthetic data generation is the Generative Adversarial Network (GAN). GAN models were first introduced by Goodfellow et al. (2014) as a tool to generate synthetic data for synthetic images. This concept was extended to a time-series synthetic generator by Yoon et al. (2019) who demonstrated its performance on stock price data, energy consumption, and health related acute lung emergencies.

Stochastic models can also be a powerful tool for synthetic data generation. For example, in generating climate data (Wilks and Wilby, 1999; Srikanthan and McMahon, 2001; Efstratiadis et al., 2014). Such models can be difficult to generalize, if they are tailored for a specific type of data. For example, precipitation might have a very different probability distribution than temperature measurements. To attempt to resolve this issue, Papalexiou (2018) presents a framework for stochastic generation of univariate and multivariate hydroclimatic data, while (Papalexiou and Serinaldi, 2020) has a similar goal with spatiotemporal random fields.

A bottom-up approach, where the analysis starts at the end node, and then is aggregated to a whole system level, has been introduced for the simulation of demand pulse distributions in water distribution systems (Buchberger and Wu, 1995). Improvements to the pulse-based approach were presented by Blokker et al. (2010) and Creaco et al. (2016). Another bottom-up approach was presented by Alvisi et al. (2014), where high-resolution data (every 1 min) per consumer were synthesized, and then aggregated over temporal and spatial axes. This approach preserved the mean, the variance, and the temporal covariance of the historical records but required detailed knowledge about the consumers to enable the high-resolution data synthesis.

When solving optimization problems related to the design and operation of complex environmental systems, the aggregated behavior is more important than the single user or end-node. For this reason, studies have also suggested top-down methods for the generation of synthetic time series. One example of a top-down approach presented by Brentan et al. (2018) used a Random-Forest algorithm for generating a random data series based on historical data combined with climatic features. Kossieris et al. (2019) suggested a stochastic approach combined with Nataf's joint distribution model to simulate water demands at different time scales. Santopietro et al. (2020) used Hermite interpolating polynomials coupled with a mixed distribution approach to generate a random data series that met known values of the daily trend.

Clearly, there are many existing methods for synthetic signal generation. However, these methods do not necessarily preserve the mean, standard deviation, and autocorrelation of the original series. Often they are domain specific, especially when data characteristics such as stationarity, seasonality, sampling scale, or series length need to be accounted for. Additionally, in many of the above methods a significant amount of data is required as input.

Here, we present a simple, easy to implement, Fourier based synthetic time series generator, that preserves the signal's first two statistical moments and the autocorrelation function. It can take any signal as input, from any domain, and it allows the user to select the level of similarity, in terms of dynamic time wrapping or Wasserstein distance (Serrà and Arcos, 2014), between the original time series and the synthesized signals. The discrete Fourier transform is a widely used mathematical tool, and efficient algorithms to execute it exist in widespread programming languages. This makes the generation method presented here quicker than many of the alternatives, which can be especially meaningful when working with large datasets, or producing a large batch of samples.

2. Methodology

2.1. Synthetic signal generator

Let $S = \{S_k\}_{k=0}^{N-1}$ be a discrete time-series signal with length N, mean μ_S, and standard deviation σ_S. The Discrete Fourier Transform (DFT) of S, denoted ζ_ω for $\omega = 0, \ldots, N-1$ is given by:

$$\zeta_\omega = \mathcal{F}\{S\}_\omega := \sum_{k=0}^{N-1} S_k e^{-i\frac{2\pi k}{N}\omega} \tag{1}$$

Since ζ_ω is a complex signal, we can use a polar representation:

$$\zeta_\omega = \rho_\omega e^{i\theta_\omega} \tag{2}$$

Where $0 \leq \rho_\omega \in \mathbb{R}$ is the amplitude and $\theta_\omega \in \mathbb{R}$ is the phase.

The generation of a synthetic signal $\hat{\zeta}_\omega$ with statistical properties of a given base signal involves the manipulation of the polar representation. Specifically, it consists of replacing the phase terms θ_ω with random phase values $\hat{\theta}_\omega$. This makes it possible to get a new random time series and keep the original signal energy because the amplitudes of the decomposed signals remain unchanged. To guarantee the conservation of the mean, the zero phase is kept unchanged, $\hat{\theta}_0 = \theta_0$. In addition to this phase, one may choose to keep $0 < m \leq N$ of the original phases, meaning $\hat{\theta}_j = \theta_j$ for any $0 < j \leq m$. This parameter allows for adjusting the synthetic signal's similarity to the original signal, as is detailed below.

After randomizing the phase components, a synthetic signal, $\hat{S}_k$, is generated by applying the inverse DFT:

$$\hat{S}_k = \mathcal{F}^{-1}\{\hat{\zeta}\}_k := \frac{1}{N} \sum_{\omega=0}^{N-1} \hat{\zeta}_\omega e^{i\frac{2\pi\omega}{N}k} \tag{3}$$

This gives us a synthetic time series $\hat{S}$ of the same length, sampling rate, and the first two statistical moments as the original time series.

One issue with this method is that the inverse transform is not guaranteed to consist only of real values. To rectify this we exploit the mathematical properties of the DFT of a real signal, so we can ensure that the synthetic signal is also real. To this end, let us consider a real signal $\{S_k\} \subset \mathbb{R}$. Its DFT holds the following conjugate symmetry:

$$\zeta_\omega = \overline{\zeta}_{N-\omega}, \qquad 1 \leq \omega \leq N-1 \tag{4}$$

This is equivalent to the following choices of $\hat{\theta}$, providing that we identify $-\pi$ with π:

$$\hat{\theta}_{N-\omega} = -\hat{\theta}_\omega, \qquad 1 \leq \omega \leq N-1 \tag{5}$$

Note that if N is even, the conjugate symmetry has a special implication for $\zeta_{N/2}$:

$$\zeta_{N/2} = \bar{\zeta}_{N-N/2} = \bar{\zeta}_{N/2} \tag{6}$$

And it is it's own conjugate, meaning $\zeta_{N/2} \in \mathbb{R}$. Thus, in order to ensure that $\hat{S}$ is real, after the randomization process in the Fourier domain, $\hat{\zeta}_{N/2}$ must fulfill:

$$\hat{\zeta}_{N/2} = \pm \zeta_{N/2} \tag{7}$$

Practically speaking, the suggested methodology is easy to implement, as many libraries implement DFT and its inverse. The constraints above on the phase randomization process ensures that the inverse transform will produce a real series. The only element that requires special consideration, in the case of a time series with an even number of elements, is $\hat{\zeta}_{N/2}$.

Thus, for a given number of phases $0 < m \le \frac{N}{2}$ the formulation for phase randomization is as follows. For the even N case:

$$\begin{cases} \hat{\theta}_\omega = \theta_\omega, & \omega = 0, \ldots, m \\ \hat{\theta}_\omega = rand(-\pi, \pi], & \omega = m+1, \ldots, \frac{N}{2}-1 \\ \hat{\theta}_{N/2} = 0 \text{ or } \pi \end{cases} \tag{8}$$

And for the odd case:

$$\begin{cases} \hat{\theta}_\omega = \theta_\omega, & \omega = 0, \ldots, m \\ \hat{\theta}_\omega = rand(-\pi, \pi], & \omega = m+1, \ldots, \frac{N-1}{2} \end{cases} \tag{9}$$

The parameter m defines how many of the first components will preserve their phase, allowing to control the similarity between the original signal S and the synthetic signal $\hat{S}$. As we increase the number of phases that are not randomized, we increase the similarity of the decomposed periodic components until we reach the point where the two signals are identical in the case of $m = \frac{N}{2}$. This capability of controlling the similarity level of the synthetic signal and the original one, while preserving the signal's statistical moments, is unique to this method.

A special case to consider is a synthetic series that is inherently non-negative, but with typical values that are close to zero. Using the randomization process described above is likely to generate synthetic series with negative values, which might not be acceptable for certain cases. As a way to amend this issue, the following optional procedure is offered. Taking the synthetic series $\hat{S}$ as our original series, a new synthetic series $\tilde{\hat{S}}$ is generated using the same method. Any negative values of $\hat{S}$ are replaced with the corresponding values of $\tilde{\hat{S}}$ (that are not guaranteed to be non-negative). If any negative values remain in $\hat{S}$, this process is repeated and a new $\tilde{\hat{S}}$ is generated. This is iterated until all values in $\hat{S}$ are non-negative. It is important to note that this process alters the statistical properties of $\hat{S}$, and while using it the mean and STD are not conserved.

2.2. Proofs of conservation of mean, standard deviation, and auto-correlation

Lemma 1. *The mean of S is equal to the mean of $\hat{S}$, $\mu_S = \mu_{\hat{S}}$.*

Proof. From the definition of $\hat{S}$:

$$\hat{S}_k = \mathcal{F}^{-1}\{\zeta_\omega\} = \frac{1}{N} \sum_{\omega=0}^{N-1} \zeta_\omega e^{-i\frac{2\pi}{N}k\omega}$$

Plugging it into the definition of the mean:

$$\mu_{\hat{S}} = \frac{1}{N} \sum_{k=0}^{N-1} \hat{S}_k = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{\omega=0}^{N-1} \zeta_\omega e^{-i\frac{2\pi}{N}k\omega} = \frac{1}{N^2} \left(N \cdot \zeta_0 + \sum_{k=0}^{N-1} \sum_{\omega=1}^{N-1} \zeta_\omega e^{-i\frac{2\pi}{N}k\omega} \right)$$

We shall now show that the sum $\sum_{k=0}^{N-1} \sum_{\omega=1}^{N-1} \zeta_\omega e^{-i\frac{2\pi}{N}k\omega}$ vanishes. Since this is a finite sum, we can flip the order of summation, and then use

the formula for a sum of a geometric series:

$$\sum_{k=0}^{N-1} \sum_{\omega=1}^{N-1} \zeta_\omega e^{-i\frac{2\pi}{N}k\omega} = \sum_{\omega=1}^{N-1} \zeta_\omega \sum_{k=0}^{N-1} e^{-i\frac{2\pi}{N}k\omega} = \sum_{\omega=1}^{N-1} \zeta_\omega \cdot \frac{\overbrace{1 - e^{-i\frac{2\pi}{N}\cdot N\omega}}^{=1}}{1 - e^{-i\frac{2\pi}{N}\omega}} = 0$$

Since the first transform coefficient is retained, $\hat{\zeta}_0 = \zeta_0$, we get:

$$\hat{\zeta}_0 = \zeta_0 = \sum_{k=0}^{N-1} S_k \underbrace{e^{-i\frac{2\pi}{N}k\cdot 0}}_{=1} = \sum_{k=0}^{N-1} S_k$$

And so:

$$\mu_{\hat{S}} = \frac{1}{N} \cdot \hat{\zeta}_0 = \frac{1}{N} \sum_{k=0}^{N-1} S_k = \mu_S \quad \square$$

Lemma 2. *The standard deviation of S is equal to the standard deviation of $\hat{S}$, $\sigma_S = \sigma_{\hat{S}}$.*

Proof. Writing out the definition for discrete standard deviation, and expanding, we get:

$$\sigma_{\hat{S}}^2 = \frac{1}{N} \sum_{k=0}^{N-1} (\hat{S}_k - \mu_{\hat{S}})^2 = \frac{1}{N} \sum_{k=0}^{N-1} (\hat{S}_k^2 - 2\hat{S}_k \mu_{\hat{S}} + \mu_{\hat{S}}^2)$$

$$= \sum_{k=0}^{N-1} \hat{S}_k^2 - 2\mu_{\hat{S}}^2 + \mu_{\hat{S}}^2 = \frac{1}{N} \sum_{k=0}^{N-1} |\hat{S}_k|^2 - \mu_{\hat{S}}^2$$

From Lemma 1, we know $\mu_S = \mu_{\hat{S}}$. From Parseval's Theorem, we get:

$$\sum_{k=0}^{N-1} |\hat{S}_k|^2 = \frac{1}{N} \sum_{\omega=0}^{N-1} |\hat{\zeta}_\omega|^2$$

Since the amplitudes are not changed during the process, $\rho_\omega = \hat{\rho}_\omega$, we get:

$$|\hat{\zeta}_\omega|^2 = |\zeta_\omega|^2$$

And so applying the identity from Parseval's theorem twice:

$$\sum_{k=0}^{N-1} |\hat{S}_k|^2 = \frac{1}{N} \sum_{\omega=0}^{N-1} |\hat{\zeta}_\omega|^2 = \frac{1}{N} \sum_{\omega=0}^{N-1} |\zeta_\omega|^2 = \sum_{k=0}^{N-1} |S_k|^2$$

And so:

$$\sigma_{\hat{S}}^2 = \frac{1}{N} \sum_{k=0}^{N-1} |\hat{S}_k|^2 - \mu_{\hat{S}}^2 = \frac{1}{N} \sum_{k=0}^{N-1} |S_k|^2 - \mu_S^2 = \sigma_S^2$$

$$\Rightarrow \sigma_{\hat{S}} = \sigma_S \quad \square$$

Lemma 3. *S and $\hat{S}$ have the same autocorrelation functions.*

Proof. As known by the Wiener–Khinchin Theorem,

$$Autocorr(S) = \mathcal{F}^{-1}\{|\zeta|^2\}$$

As mentioned before, since the amplitudes are unchanged, $|\hat{\zeta}_\omega|^2 = |\zeta_\omega|^2$, and so:

$$Autocorr(\hat{S}) = \mathcal{F}^{-1}\{|\hat{\zeta}|^2\} = \mathcal{F}^{-1}\{|\zeta|^2\} = Autocorr(S) \quad \square$$

2.3. Comparison

The above characteristics of the Fourier based method make it easy and convenient to use. However, many synthetic signal generation methods have been presented in the literature. Here, we use the following for comparison.

2.3.1. Auto Regressive Moving Average (ARMA)

The first method used for comparison is the Auto Regressive Moving Average (ARMA). While ARMA is typically used for time series forecasting, it can also be used for generation. It combines an autoregressive model $AR(p)$, with a moving average model $MA(q)$. Briefly, in an $ARMA(p, q)$ model a given time step y_t is calculated using the p previous time steps, and q past noise elements ε_t, as such:

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \qquad (10)$$

Where $c, \phi_{1,\ldots,p}, \theta_{1,\ldots,q}$ are the model's parameters. See Hyndman and Athanasopoulos (2021) for more details. In our implementation we used the Statsmodels python library (Seabold and Perktold, 2010) to find the model's parameters given the hyperparameters p, q, and then generate time series accordingly.

2.3.2. GAN

As mentioned before, a much newer method for signal generation is Generative Adversarial Networks (GAN). The general idea is to place two neural networks in a zero sum game. The first model, the aptly named Generator, is tasked with generating data. The second model, the Discriminator, is attempting to distinguish between the synthetic data and the original data set. The models are trained side by side, with the generator training to "fool" the discriminator, which in turn trains to improve at discerning fake data. A trained model can generate a lot of synthetic data relatively quickly, and if the process converged properly it should resemble the original dataset. It is important to note that a major characteristic of GAN is that it acts as black box — no direct control of the generator model is possible before or after training. The architecture used for comparison is quantGAN (Wiese et al., 2020), designed for modeling financial time series. Although intended for a different field, it fits our use case. Using temporal convolutional networks, it can capture both the distribution of the data, and the autocorrelation in different lags. This is in contrast to GANs intended for tabular data (Xu et al., 2019), that emulate the distribution but not the temporal dependencies. QuantGAN accepts a single series as input of any length, allowing us to train it on each dataset used for comparison. The hyperparameters for each dataset were determined by trial and error.

The main deviation from the methodology of the original quantGAN is in the way our data was scaled in pre-processing. For proper performance, the time series values fed into the GAN should be in the range $(-1, 1)$. In the original paper, the scaling is done in a method that relies on specific properties of the financial data used. To achieve this for any time series, the MinMaxScaler function from Scikit-learn (Pedregosa et al., 2011) was used, as it gave the best results of all scalers tested. The original data was transformed in pre-processing, and the generated series were transformed back to the original scale using the inverse operation.

2.3.3. CoSMoS

CoSMoS (Papalexiou et al., 2021) is an R package for stochastic generation of synthetic data, with a focus on hydroclimatic datasets. One use case is generating synthetic data from scratch, by choosing a probability distribution for the data, and an autocorrelation structure (ACS). However, it can also take an existing time series. One chooses the type of distribution appropriate for the data type (Generalized Gamma, Burr, etc.) and the type of ACS. Then the parameters of the distribution and ACS are fitted to the given time series. This allows generation of new synthetic time series with the same probability distribution and ACS as the original.

2.4. Data

A variety of environmental datasets were chosen as examples for this paper. This was done to offer varying conditions for testing the performance of the suggested method. The data consists of domestic water demand, concentration of air pollutants, wave heights, and turbulent air flow velocities. These signals differ in their length, the rate of change (*i.e.* frequency components) and the typical observed values. Some are inherently positive and fluctuate close to zero, while others change sign often or have large positive values throughout.

For each dataset, 1,000 synthetic series were generated by each method. This was done since all generation methods have a random element, so analyzing the signal's properties over a large set provides a more accurate insight.

2.4.1. Urban water demand

The urban water demand dataset contains hourly values of water demand of a metropolitan in cubic meters, taken from the connection between the metro and the national water company. The metro has about 1 million residents, and is spanning over 126 squared kilometers. Average height above sea level is 715 m. Temperature averages are 25 °C in the hottest months, July–August, and about 10° in the coldest, January. The full data spans a year, from 1/1/2020 until 31/12/2020. For clearer data visualization, only the month of July was used, giving 745 hourly time-steps.

2.4.2. Air pollution

Within the context of this study two types of air pollutants datasets have been investigated: urban ambient nitric oxide (NO) levels; and fine Particulate Matter (PM 2.5) with a diameter of $2.5[\mu m]$ or less, both in $[\frac{\mu g}{m^3}]$. The concentrations of both pollutants have been acquired by the same AQM standard monitoring station. The station was positioned 20 meters above ground, and 235 meters above sea level. The city has approximately 300,000 residents, over an area of about 63.5 squared kilometers. The data contains daily averages, from 1/1/2023 to 31/12/2023. For the sake of brevity, the PM results are detailed in the supplementary material.

2.4.3. Air flow velocity

Turbulent Air flow velocity, measured in an arbitrary location near Nofit, 18 km east of Haifa, Israel, taken 179 meters above sea level. The original files contain three orthogonal velocity field components $\vec{V} = (u(t), v(t), w(t))$, measured in $[m/s]$. The measurements sampling frequency was 2 kHz, and one minute of measurement was used. The u component was arbitrarily selected to represent this set of measurements.

2.4.4. Wave height

Another environmental domain used here is wave field synthetic data generation. To this end, wave height records from the gulf of Eilat were used (Shani-Zerbib et al., 2017, 2018). Data was collected continuously for over 50 h during June 2017, taken with an array of 5 wavegauges, positioned in a circle. Measurements are statistically identical between the gauges, with variations in phase. Sampling frequency was 80 Hz. The full set of measurements is available via Mendeley (Shani-Zerbib and Liberzon, 2018).

2.5. Comparison of synthetic time series

Quantitative methods of comparison between the generated time series are essential to understanding the differences between the generation methods. 4 statistical measures were chosen — mean, standard deviation (SD), skewness, and kurtosis. The values were calculated for each synthetic series, then averaged over all 1,000 samples.

However, statistical properties lack the ability to reflect seasonal trends, and to describe how similar are a pair of compared time series

in any specific point in time. A pair of time series may have a similar mean and SD, and still have vastly different plots and distributions. Thus additional similarity measures, specific to time series analysis were used. Here, two metrics highlighted in Brophy et al. (2023) were employed — Dynamic Time Warping and the Wasserstein Distance.

Dynamic Time Warping (DTW) measures of the distance between a pair of time series. DTW and its variants are often used for time series similarity analysis, and are considered to be a good metric for comparison (Serrà and Arcos, 2014). Given two time series $X = \{x_i\}_{i=1}^n$ and $Y = \{y_j\}_{j=1}^m$, one can create a path π between them, matching each element in X to an element of Y, and vice versa. This matching is not necessarily one-to-one, but all points should have a match. In addition, x_1 is always matched to y_1, and x_n to y_m. The matching needs to respect order (the path cannot "cross" itself). Then DTW is computed as a minimum of the Euclidean distance on all such paths:

$$DTW(x, y) = \min_{\pi} \sqrt{\sum_{(i,j) \in \pi} d\left(x_i, y_j\right)^2} \qquad (11)$$

We compare all synthetic series generated to the original using DTW, and then take the average and standard deviation (SD). The **lowest** score represents the method generating synthetic signals that are the most similar to the original series.

Wasserstein distance (WD, also known as Earth Mover distance), measures the "cost" of turning one distribution into another, giving a way to quantitatively compare distributions. Here we refer to Wasserstein-1. Given probability mass functions u, v (corresponding to time series, in our case) the formula is:

$$W_1(u, v) = \inf_{\pi \in \Gamma(u,v)} \int_{\mathbb{R} \times \mathbb{R}} |x - y| d\pi(x, y) \qquad (12)$$

Where $\Gamma(u, v)$ are the set of all joint probability measures on $\mathbb{R} \times \mathbb{R}$ with marginal distributions u, v, respectively. Here as well, we compare all synthetic series to the original, then take the average and SD of the results. The **lowest** WD score belongs to the method that generates series with distributions that are the most similar to the original distribution.

Synthetic data should maintain characteristics of the original data it is trying to replicate, so some similarity to the original data is desirable. However, if it is too similar to the original data, it limits the range of possible outcomes considered, and thus it will not be able to predict cases differing substantially from the given original sample. Hence the optimal similarity level is application dependant. The method in hand, as described above, presents a mechanism to control the similarity level.

Two other useful tools for time series analysis are the distribution function, and the Autocorrelation Function (ACF) plot. Those were also produced for a single series from each generation method for every dataset, to give more insight about the temporal dependencies.

3. Results & discussion

3.1. Control similarity

One of the key features of the suggested method is the ability to control the synthetic generated signal's similarity to the original signal. This is done through the number of phase components that are randomized (see Eqs. (8) and (9)). The effect of randomized components is illustrated in Fig. 1, where the original signal and a cohort of synthetic signals are presented for all datasets in this study. The original signal is presented in blue, and in the background a set of 100 synthetic signals are plotted in light blue. The left column in Fig. 1 depicts the original signal in front of a synthesized set, where only the first three phase components are preserved, while the rest are randomized. The second column presents the same configuration with 40 preserved phase coefficients. The right-hand column plots the original signal on top of 100 synthetic signals that have been generated with 100 preserved phase coefficients. It is clear that the more coefficients are preserved, the higher the similarity of the generated signals. This is also

illustrated in . In particular, from .B and Figs. 2–5 we can discern that the similarity between the original and the synthetic signals increases with m, as evident by the decrease in DTW and WD. Yet, for any m value, the mean and SD are preserved.

3.2. Urban water demand

For urban water demand we used $m = 3$ for the DFT method. The model parameters for ARMA were selected to be $p = 3, q = 3$, as those have provided the best results in terms of visual similarity to the original. For CoSMoS, the distribution was set to Generalized Gamma, and the ACS was Pareto II. As seen in Fig. 2, the autocorrelation of the original series contains negative correlations that cannot be obtained by Pareto II. This is the case for the included ACSs, and our attempts to create a custom ACS were unsuccessful.

Table 2 presents the mean, standard deviation (SD), skewness, kurtosis, DTW and WD measures for the urban water demand time series. Fourier retains the mean and SD, as expected. The skewness is not retained on average, but the Fourier method does produce the closest value to the skewness of the original series. Examining the SD of the skewness suggests that the individual synthetic series have similar skewness to the original, but with alternating signs. The average kurtosis of the series generated by the Fourier method, presented here, is closest to the original. Fourier also achieved the lowest DTW score, 10% lower than the next, ARMA, and 20% lower than the highest, CoSMoS. Fourier was best in WD as well, 12% lower than ARMA. Note however the large variability, as the SD is 42% of the average score.

Fig. 2 takes the original series, as well as a random synthetic series from each of the generation methods, and plots them together for comparison. A histogram is plotted, using the same bins for all the plots for easier assessment. The autocorrelation function (ACF) is plotted up to 10 lags. Looking at Fig. 2, GAN has a similar autocorrelation shape, and the plot is visually similar to the original. However, GAN received the lowest WD score, 3.8 times larger than Fourier, and the second-worst DTW score. This disparity might be explained by the average mean, that is 6% lower than the original series. Numerous attempts to train the GAN did not manage to achieve a closer mean value. CoSMoS got a mean WD score of 275, implying greater distribution similarity, compared to GAN. However, it's DTW score is the worst, and the example plot in Fig. 2 displays erratic behavior not found in the original. This can be explained by the autocorrelation structure.

3.3. Air pollution

Clearly this data should have non-negative values only, yet the typical values were close to zero. For this reason the non-negative implementation described in Section 2.1 was used here. For Fourier generation, $m = 100$ is used. A much higher number of coefficients was kept here, since as visible in Fig. 1, even with $m = 100$ the synthetic series have a decent amount of deviation from the original. For ARMA, $p = 3, q = 3$, and in CoSMoS Generalized Gamma dist. was used, with Weibull ACS.

Table 3 presents the statistical comparison between the methods for the NO data. NO data was characterized by small values around zero, while the time series physically cannot produce negative values. Therefore the non-negative procedure was applied. That is why the Fourier methodology did not conserve the mean and SD in this dataset. In fact both the mean and SD stray the furthest from the original, with deviations of 11.8% and 11.4%, respectively. ARMA reproduced the mean and SD. CoSMoS has an avg. mean notably lower than the original, with 6.5% difference, while it did reproduce the closest skewness values to the original. The kurtosis is especially notable here as it is orders of magnitude greater in this dataset, compared to the others. CoSMoS did the best job on average reproducing the kurtsis. 31% difference from the original, compared to 106% in the next best (GAN). However, the SD is larger than the value itself, indicating large

Table 1
Statistics and metrics averaged over 100 samples for each specified *m* value, using the Fourier method.

Table 1.A Mean, Standard Deviation (SD), Skewness, and Kurtosis for each *m* value.

	Mean	SD	Skewness	Kurtosis
Water Demand [m³]	14 528	1915	−0.26	−0.34
m = 3	14528 ± 0	1916 ± 0	−0.02 ± 0.31	−0.43 ± 0.25
m = 40	14528 ± 0	1916 ± 0	−0.09 ± 0.28	−0.20 ± 0.33
m = 100	14528 ± 0	1916 ± 0	−0.25 ± 0	−0.35 ± 0.02
NO [μg/m³]	0.64	0.65	4.58	34.25
m = 3	0.75 ± 0.01	0.52 ± 0.01	0.71 ± 0.12	0.08 ± 0.41
m = 40	0.74 ± 0.01	0.54 ± 0.01	0.92 ± 0.13	0.77 ± 0.6
m = 100	0.7 ± 0.01	0.59 ± 0.01	1.97 ± 0.17	6.8 ± 1.5
u [m/s]	1.47	0.33	0.003	−0.56
m = 3	1.47 ± 0	0.33 ± 0	−0.18 ± 0.22	−0.50 ± 0.35
m = 40	1.47 ± 0	0.33 ± 0	0.02 ± 0.02	−0.55 ± 0.06
m = 100	1.47 ± 0	0.33 ± 0	0.02 ± 0.01	−0.61 ± 0.01
Wave Height [m]	−0.0092	0.04	−0.17	−0.63
m = 3	−0.0092 ± 0	0.04 ± 0	−0.009 ± 0.272	−0.273 ± 0.414
m = 40	−0.0092 ± 0	0.04 ± 0	−0.182 ± 0.007	−0.576 ± 0.026
m = 100	−0.0092 ± 0	0.04 ± 0	−0.172 ± 0.001	−0.621 ± 0.002

Table 1.B Metric values for both Dynamic Time Warping (DTW) and Wasserstein Distance (WD), quantifying the similarity of the synthetic samples to the original time series for each *m* value. Lower scores indicate greater similarity.

	DTW	WD
Water Demand [m³]	–	–
m = 3	21866 ± 1868	230 ± 98
m = 40	13033 ± 1766	239 ± 104
m = 100	3621 ± 216	28 ± 2
NO [μg/m³]	–	–
m = 3	9.2 ± 0.21	0.19 ± 0.01
m = 40	8.78 ± 0.23	0.17 ± 0.01
m = 100	7.54 ± 0.38	0.11 ± 0.01
u [m/s]	–	–
m = 3	5.41 ± 0.81	0.03 ± 0.01
m = 40	2.63 ± 0.15	0.01 ± 0.005
m = 100	1.65 ± 0.05	0.01 ± 0.005
Wave Height [m]	–	–
m = 3	0.687 ± 0.109	0.005 ± 0.002
m = 40	0.171 ± 0.001	0.001 ± 0.0001
m = 100	0.111 ± 0.003	0.0004 ± 0.0001

Table 2
Statistics and metrics for Urban Demand Dataset. Each value is the average of all 1,000 synthetic series produced with the same method, ± SD. In bold are the scores indicating greatest similarity to the original series, meaning the closest mean, SD, Skewness and Kurtosis, and the lowest DTW and WD scores.

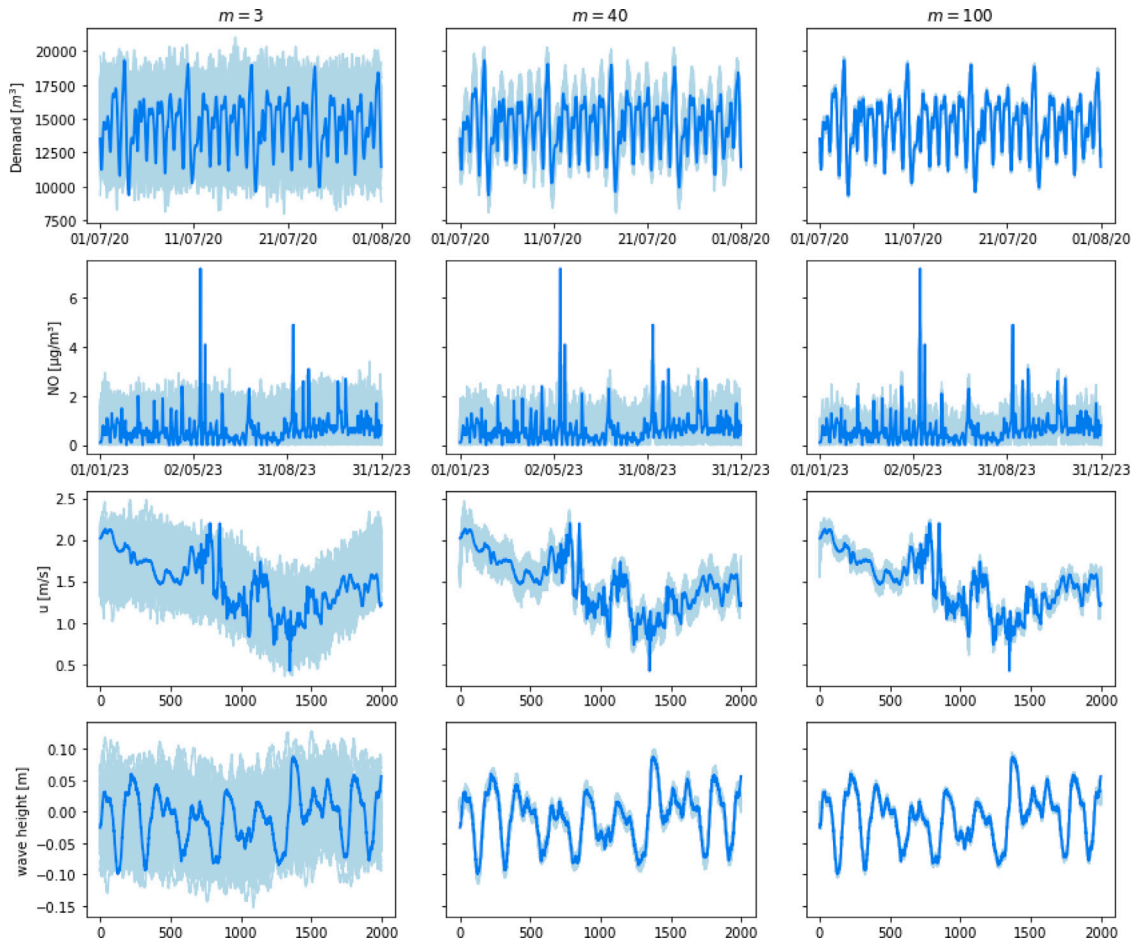| m=3	Original	Fourier	ARMA	GAN	CoSMoS
Mean	14 528	**14528** ± 0	14536 ± 160	13645 ± 41	14526 ± 217
SD	1916	**1916** ± 0	1909 ± 123	1844 ± 161	1881 ± 108
Skewness	−0.26	0 ± **0.31**	−0.00 ± 0.19	−0.08 ± 0.17	−0.04 ± 0.19
Kurtosis	−0.34	**−0.43** ± 0.26	−0.05 ± 0.32	0.11 ± 0.31	0.16 ± 0.45
DTW	–	**22087** ± 1942	24518 ± 1783	26866 ± 1343	26894 ± 1796
WD	–	**236** ± 100	266 ± 62	894 ± 42	275 ± 82

Fig. 1. Comparing different m values for each dataset.

Table 3

Statistics and metrics for NO dataset. Average values of the synthetic series $\pm$ SD. Closest statistics and lowest metric scores in bold.

m = 100	Original	Fourier	ARMA	GAN	CoSMoS
Mean	0.64	0.72 ± 0.01	**0.64** ± 0.05	0.67 ± 0.01	0.60 ± 0.04
SD	0.65	0.58 ± 0.01	**0.65** ± 0.03	0.64 ± 0.06	0.62 ± 0.13
Skewness	4.58	1.99 ± 0.17	0.00 ± 0.13	2.83 ± 0.30	**3.52** ± 1.86
Kurtosis	34.25	7.09 ± 1.56	−0.03 ± 0.24	10.57 ± 2.82	**24.96** ± 29.32
DTW	–	**7.49** ± 0.37	10.65 ± 0.35	8.84 ± 0.63	7.93 ± 1.15
WD	–	0.12 ± 0.01	0.26 ± 0.02	0.085 ± 0.011	**0.08** ± 0.017

differences in kurtosis between samples. Fourier has the lowest DTW, but only 5.7% lower than CoSMoS. CoSMoS got the lowest WD score, slightly lower than GAN.

A different approach for obtaining non-negative was tested. A synthetic signal was generated, and then any negative values were simply set to zero. This approach, with $m = 100$, yielded a mean of 0.7 ± 0.07, SD of 0.586 ± 0.009, skewness of 2 ± 0.2, kurtosis of 6.8 ± 1.4, DTW score of 7.5 ± 1.4, and WD score of 0.11 ± 0.007. The averages were close to the original non-negative implementation used. This indicates that this alternative might be also viable in cases with few negative values.

Despite the fact ARMA reproduced the mean and SD, it has the highest scores in both metrics, as well as averaging a negative kurtosis value. The example series for ARMA in Fig. 3 shows negative values are obtained, which is undesirable in this case. The other methods show

similarity to the original, with non-negative values in the range $[0, 2]$ and occasional sharp peaks. This is also visible in the distribution.

3.4. Air flow velocity

$m = 5$ is used in Fourier, chosen by trial and error. For ARMA, $p = 2, q = 4$, and in CoSMoS Burr type XII was used for both the distribution and the ACS.

Looking at Table 4, all methods managed to reproduce the mean with up to 0.01 difference from the original series. Fourier was the only method to reproduce the SD, with CoSMoS coming the closest, with a 10% lower average. With regards to skewness, ARMA did best. However, the SD of the skewness was 16 times larger than the average value, indicating big variations between different generated series. GAN
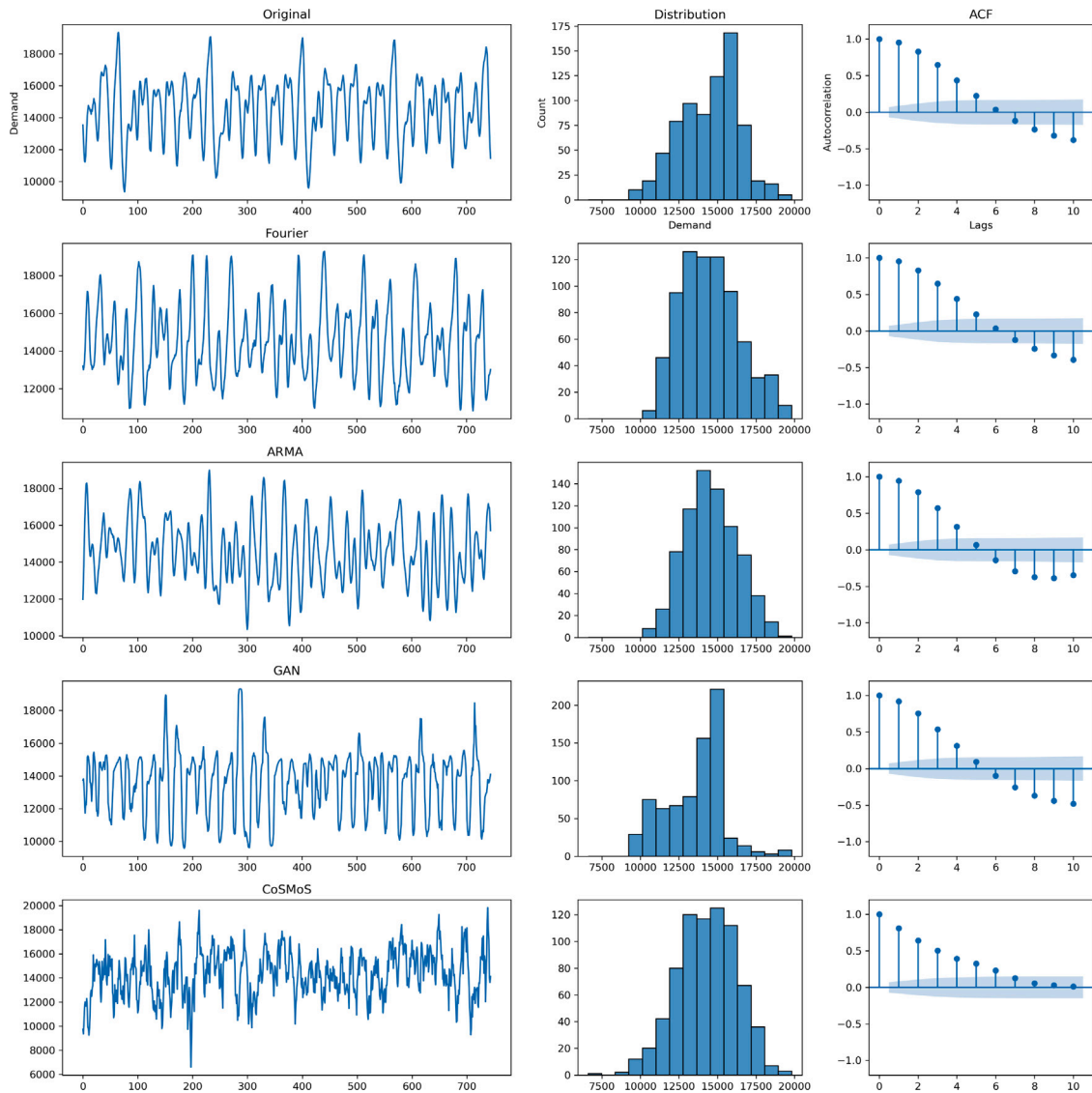
Fig. 2. The original time series (at the top) and a single time series produced by each method for the Urban Demand dataset, including distribution and ACF for each. Colored area in ACF depicts the confidence interval of 95%.

did best with replicating kurtosis, followed by Fourier. Fourier again scored lowest on both DTW and WD, with CoSMoS being the closest in both. Average DTW for CoSMoS was 11% more than Fourier, and WD was 103% greater. GAN and ARMA got similarly high scores in both metrics. The GAN example series in Fig. 4 displays erratic behavior, and ACF values that oscillate rather than decrease monotonically (as in the original). The ARMA sample seems realistic, in terms of the range of values and the distribution and ACF plots. The thing it does not seem to replicate is the general trend of the series itself — in indices up to 1,000 the original signal gets values in the range [1, 2.25], and after it the series presents lower values, in the range of [0.5, 1.75]. The CoSMoS and Fourier samples show similar behavior, but ARMA does not, which might explain the relatively high scores it got from the metrics.

3.5. Wave height

Here, $m = 5$ is used. For ARMA, $p = 6, q = 10$, and in CoSMoS Generalized Gamma is used for the distribution, with Burr type XII for the ACS.

Table 5 details the results for the wave height dataset. As expected, Fourier reproduced the mean and SD. ARMA came the closest with only 2.2% difference, and GAN the furthest, getting an average mean 26%

smaller than the original. ARMA did the best in terms of reproducing skewness, while CoSMoS did best in kurtosis, with large variability in both. Fourier got the lowest scores in both metrics once again, with ARMA second lowest in both. DTW was 24.3% higher in ARMA, while WD was 46.5% higher.

Fig. 5 gives a similar assessment as to the performance of the methods. Fourier and ARMA have samples closely resembling the original. In GAN, the general trends and the distribution appear similar, but the series values change much more sharply than in the original. Also, similar to the u and PM 2.5 GAN samples, the ACF oscillates in a way not present in the original, at least not to a visible degree. Note how the CoSMoS sample receives a much smaller range of values than the original and others, as evident in the distribution plot.

4. Conclusions

In 4 out of 5 datasets, Fourier did a superior job in reproducing both the mean and SD, and got the lowest WD scores. It got the lowest DTW score in all 5. This shows the capability of this method to take a single sample and produce thousands of different synthetic time series with similar statistical properties, visual shape, autocorrelation, and distribution. The ability of the method to set the similarity level
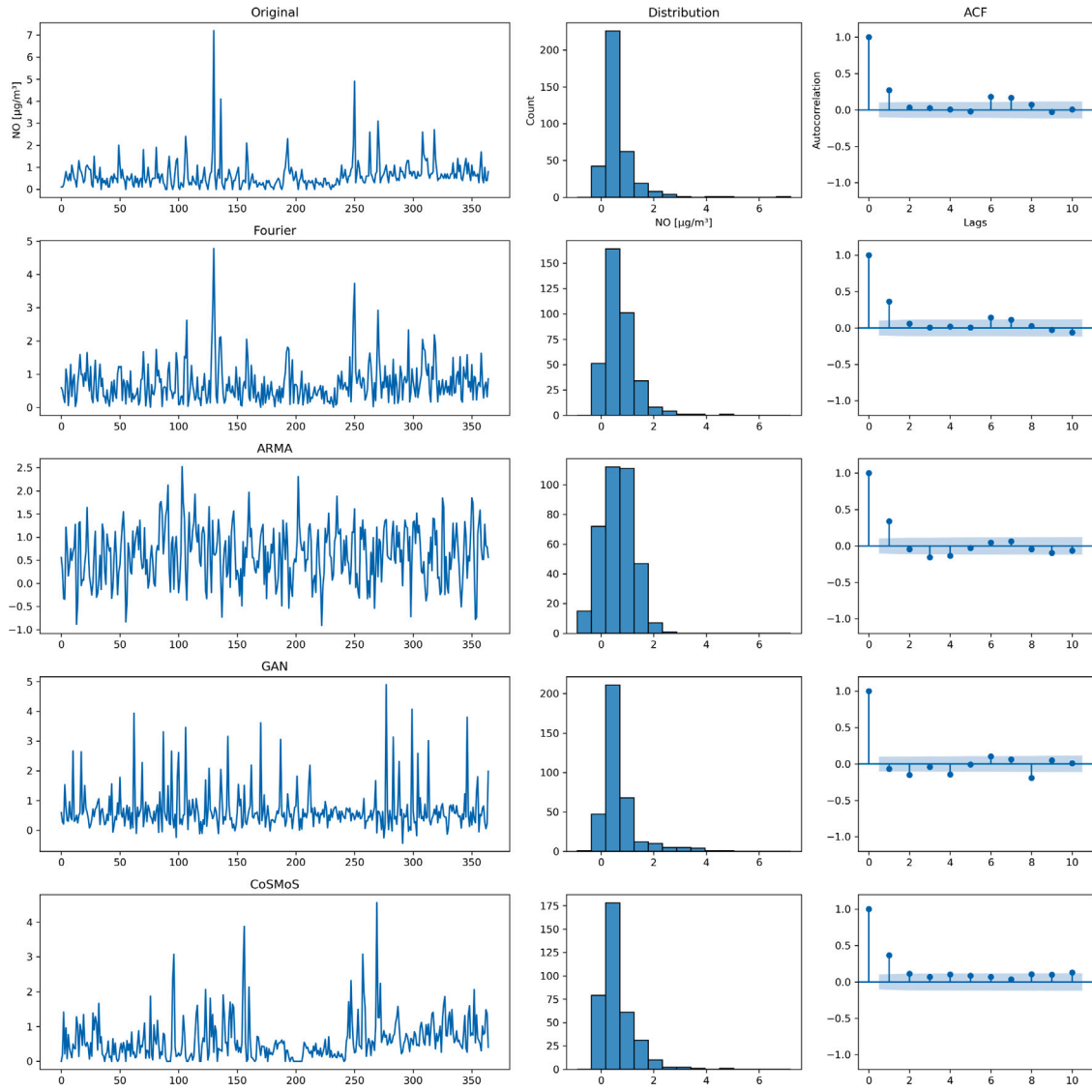
Fig. 3. NO — original time series and a single time series produced by each method, including distribution and ACF.

Table 4
Statistics and metrics for the u dataset (turbulent air flow velocity component). Average values of the synthetic series $\pm$ SD. Closest statistics and lowest metric scores in bold.

m=5	Original	Fourier	ARMA	GAN	CoSMoS
Mean [m/s]	1.47	**1.47** $\pm$ 0	**1.47** $\pm$ 0.09	1.46 $\pm$ 0.06	1.46 $\pm$ 0.06
SD [m/s]	0.33	**0.33** $\pm$ 0	0.28 $\pm$ 0.04	0.26 $\pm$ 0.03	0.30 $\pm$ 0.04
skewness	0.003	0.07 $\pm$ 0.16	**0.02** $\pm$ 0.32	−0.09 $\pm$ 0.24	−0.25 $\pm$ 0.31
Kurtosis	−0.56	−0.45 $\pm$ 0.26	−0.18 $\pm$ 0.47	**−0.63** $\pm$ 0.31	−0.41 $\pm$ 0.65
DTW	–	**4.69** $\pm$ 0.47	8.79 $\pm$ 2.66	8.90 $\pm$ 2.27	5.24 $\pm$ 1.04
WD	–	**0.023** $\pm$ 0.005	0.089 $\pm$ 0.04	0.077 $\pm$ 0.03	0.072 $\pm$ 0.03

Table 5
Statistics and metrics for the wave height dataset $\pm$ SD. Closest statistics and lowest metric scores in bold.

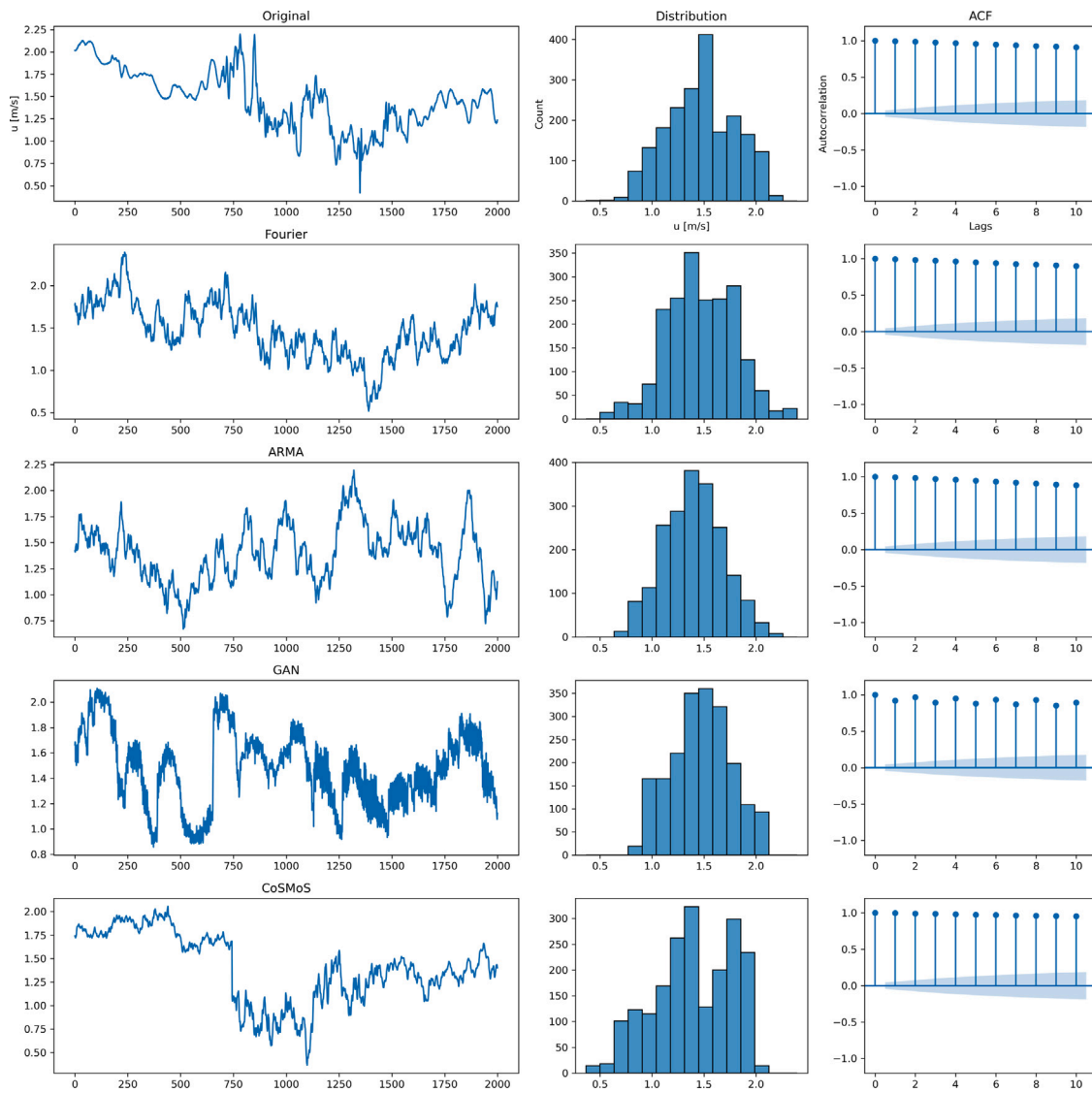| m=5	Original	Fourier	ARMA	GAN	CoSMoS
Mean	−0.0092	**−0.0092** $\pm$ 0	−0.009 $\pm$ 0.006	−0.012 $\pm$ 0.006	−0.01 $\pm$ 0.018
SD	0.04	**0.0404** $\pm$ 0	0.041 $\pm$ 0.005	0.047 $\pm$ 0.004	0.031 $\pm$ 0.007
skewness	−0.17	**0.009** $\pm$ 0.27	−0.0029 $\pm$ 0.31	−0.079 $\pm$ 0.185	−0.043 $\pm$ 0.43
Kurtosis	−0.63	−0.26 $\pm$ 0.42	−0.15 $\pm$ 0.51	−0.95 $\pm$ 0.22	**−0.35** $\pm$ 0.58
DTW	–	**0.65** $\pm$ 0.1	0.83 $\pm$ 0.13	0.86 $\pm$ 0.12	1.12 $\pm$ 0.14
WD	–	**0.005** $\pm$ 0.002	0.008 $\pm$ 0.003	0.009 $\pm$ 0.004	0.018 $\pm$ 0.009

Fig. 4. *u* - The original time series (at the top) and a single time series produced by each method.

allows for a great flexibility that is not available in all other methods. The method is also simple to implement and computationally efficient, which make it scalable. The various examples clearly show that the method is also versatile, and robust to the temporal resolution of the original signal.

A straightforward direction for future research is to expand this idea to higher dimensions, using the multi-dimensional Fourier transform. Another possibility is to use this method for data augmentation of a machine-learning algorithm - *i.e.* to use this method to generate synthetic series for training a GAN. Finally, this method currently uses only a single instance of time series data. If, for example, the same set of data from different years is available, this method may be further developed so it could leverage this additional information to produce a wider range of synthetic series.

CRediT authorship contribution statement

Ofek Aloni: Writing – original draft, Validation, Software, Investigation. **Gal Perelman:** Writing – review & editing, Software. **Barak Fishbain:** Writing – review & editing, Supervision, Project administration, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Barak Fishbain, Ofek Aloni reports financial support was provided by Israel Water Authority. Barak Fishbain, Ofek Aloni reports financial support was provided by Israeli Ministry of Science and Technology. Prof. Dan Liberzon and his lab, the Technion Sea-Atmosphere Interactions Research Laboratory, have provided the waves height data. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
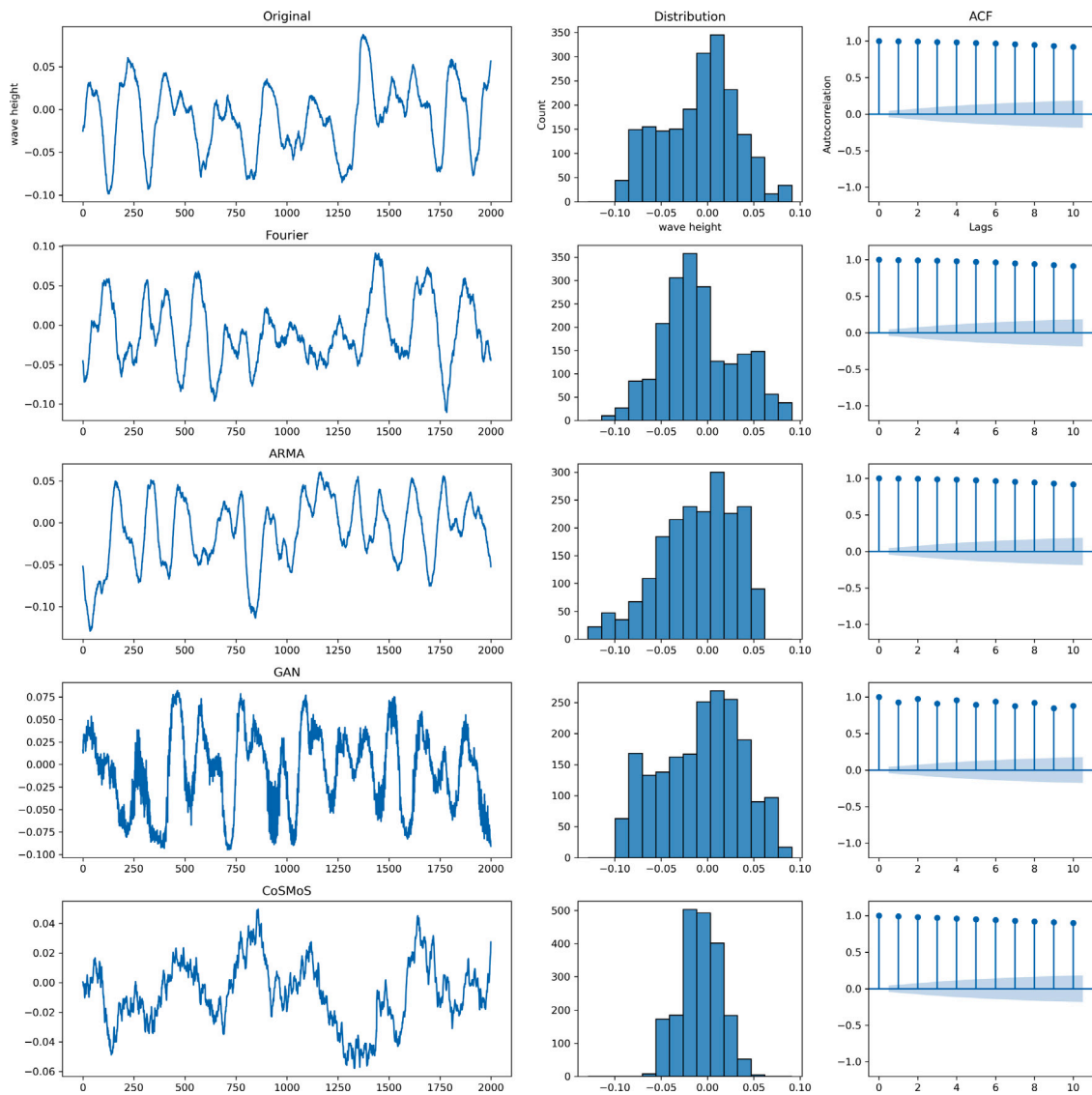
Acknowledgments

Fig. 5. Wave Height — The original time series (at the top) and a single time series produced by each method.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.envsoft.2024.106283.

Software and data availability

• Name of software: **Random Environmental Time Series Generator**

• Developer: Ofek Aloni
• Contact: ofek.aloni@campus.technion.ac.il
• Date first available: Oct. 15th, 2024.
• Program language: Python
• Source code at: https://github.com/Al-Ofek/stsg

• Documentation: Detailed documentation for application, data, testing, and deployment can be found at https://github.com/Al-Ofek/stsg/blob/main/README.md
• Data required for local installation and use of software can be found under https://github.com/Al-Ofek/stsg.
• Software Required: **QuantGAN** model was implemented using Pytorch. Since training a GAN is computationally intensive, a virtual machine was used for training. The file used for training is included, as well as a variety of generator models pre-trained on the used datasets, and can be found under "GAN Generator" subdirectory. For this reason the work files only import a model, then use it for generating synthetic series. **CoSMoS** is available only as an R package, the synthetic data was generated using R, then imported into Python. Working files can be found in the cosmos directory. For more on CoSMoS, see https://CRAN.R-project.org/package=CoSMoS.

References

Alvisi, S., Ansaloni, N., Franchini, M., 2014. Generation of synthetic water demand time series at different temporal and spatial aggregation levels. Urban Water J. 11 (4), 297–310. http://dx.doi.org/10.1080/1573062X.2013.801499.

Blokker, E.J.M., Vreeburg, J.H.G., Van Dijk, J.C., 2010. Simulating residential water demand with a stochastic end-use model. J. Water Resour. Plan. Manag 136 (1), 19–26. http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000002.

Box, G.E.P., Jenkins, G., 1990. Time Series Analysis, Forecasting and Control. Holden-Day, Inc., USA.

Brentan, B.M., Meirelles, G.L., Manzi, D., Luvizotto, E., 2018. Water demand time series generation for distribution network modeling and water demand forecasting. Urban Water J. 15 (2), 150–158. http://dx.doi.org/10.1080/1573062X.2018.1424211.

Brophy, E., Wang, Z., She, Q., Ward, T., 2023. Generative adversarial networks in time series: A systematic literature review. ACM Comput. Surv. 55 (10), 199:1–199:31. http://dx.doi.org/10.1145/3559540.

Buchberger, S.G., Wu, L., 1995. Model for instantaneous residential water demands. J. Hydraul. Eng. 121 (3), 232–246. http://dx.doi.org/10.1061/(ASCE)0733-9429(1995)121:3(232).

Chen, Y., Song, L., Liu, Y., Yang, L., Li, D., 2020. A review of the artificial neural network models for water quality prediction. Appl. Sci. 10 (17), 5776. http://dx.doi.org/10.3390/app10175776.

Creaco, E., Kossieris, P., Vamvakeridou-Lyroudia, L., Makropoulos, C., Kapelan, Z., Savic, D., 2016. Parameterizing residential water demand pulse models through smart meter readings. Environ. Model. Softw. 80, 33–40. http://dx.doi.org/10.1016/j.envsoft.2016.02.019.

Efron, B., 1992. Bootstrap methods: Another look at the Jackknife. In: Kotz, S., Johnson, N.L. (Eds.), Breakthroughs in Statistics. Springer New York, New York, NY, pp. 569–593.

Efstratiadis, A., Dialynas, Y.G., Kozanis, S., Koutsoyiannis, D., 2014. A multivariate stochastic model for the generation of synthetic time series at multiple time scales reproducing long-term persistence. Environ. Model. Softw. 62, 139–152. http://dx.doi.org/10.1016/j.envsoft.2014.08.017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in Neural Information Processing Systems. 27, Curran Associates, Inc..

Herman, J.D., Zeff, H.B., Lamontagne, J.R., Reed, P.M., Characklis, G.W., 2016. Synthetic drought scenario generation to support bottom-up water supply vulnerability assessments. J. Water Resour. Plan. Manag 142 (11), 04016050. http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000701.

Hill, M.J., Roxburgh, S.H., Carter, J.O., Barrett, D.J., 2006. Development of a synthetic record of fire probability and proportion of late fires from simulated growth of ground stratum and annual rainfall in the Australian tropical Savanna zone. Environ. Model. Softw. 21 (8), 1214–1229. http://dx.doi.org/10.1016/j.envsoft.2005.06.008.

Hirsch, R.M., Archfield, S.A., De Cicco, L.A., 2015. A bootstrap method for estimating uncertainty of water quality trends. Environ. Model. Softw. 73, 148–166.

Hyndman, R.J., Athanasopoulos, G., 2021. Forecasting: Principles and Practice. 3rd Edition. Melbourne, Australia, OTexts.

Kang, D., Lansey, K., 2014. Multiperiod planning of water supply infrastructure based on scenario analysis. J. Water Resour. Plan. Manag 140 (1), 40–54. http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000310.

Kilsby, C., Jones, P., Burton, A., Ford, A., Fowler, H., Harpham, C., James, P., Smith, A., Wilby, R., 2007. A daily weather generator for use in climate change studies. Environ. Model. Softw. 22 (12), 1705–1719.

Kofinas, D.T., Spyropoulou, A., Laspidou, C.S., 2018. A methodology for synthetic household water consumption data generation. Environ. Model. Softw. 100, 48–66. http://dx.doi.org/10.1016/j.envsoft.2017.11.021.

Kossieris, P., Tsoukalas, I., Makropoulos, C., Savic, D., 2019. Simulating marginal and dependence behaviour of water demand processes at any fine time scale. Water 11 (5), 885. http://dx.doi.org/10.3390/w11050885.

McPhail, C., Maier, H., Westra, S., Van Der Linden, L., Kwakkel, J., 2021. Guidance framework and software for understanding and achieving system robustness. Environ. Model. Softw. 142, 105059. http://dx.doi.org/10.1016/j.envsoft.2021.105059.

Moshenberg, S., Lerner, U., Fishbain, B., 2015. Spectral methods for imputation of missing air quality data. Environ. Syst. Res 4 (1), 26. http://dx.doi.org/10.1186/s40068-015-0052-z.

Negra, N.B., Holmstrøm, O., Bak-Jensen, B., Sørensen, P., 2008. Model of a synthetic wind speed time series generator. Wind Energy 11 (2), 193–209. http://dx.doi.org/10.1002/we.244.

Nowak, K., Prairie, J., Rajagopalan, B., Lall, U., 2010. A nonparametric stochastic approach for multisite disaggregation of annual to daily streamflow. Water Resour. Res. 46 (8), http://dx.doi.org/10.1029/2009WR008530, 2009WR008530.

Odan, F.K., Reis, L.F.R., 2012. Hybrid water demand forecasting model associating artificial neural network with Fourier series. J. Water Resour. Plan. Manag 138 (3), 245–256. http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000177.

Papalexiou, S.M., 2018. Unified theory for stochastic modelling of hydroclimatic processes: Preserving marginal distributions, correlation structures, and intermittency. Adv. Water Resour. 115, 234–252. http://dx.doi.org/10.1016/j.advwatres.2018.02.013.

Papalexiou, S.M., Serinaldi, F., 2020. Random fields simplified: Preserving marginal distributions, correlations, and intermittency, with applications from rainfall to humidity. Water Resour. Res. 56 (2), http://dx.doi.org/10.1029/2019WR026331, e2019WR026331.

Papalexiou, S.M., Serinaldi, F., Strnad, F., Markonis, Y., Shook, K., 2021. CoSMoS: complete stochastic modelling solution. https://CRAN.R-project.org/package=CoSMoS.

Papamichail, D.M., Georgiou, P.E., 2001. Seasonal ARIMA inflow models for reservoir sizing. J. Am. Water Resour. Assoc. 37 (4), 877–885. http://dx.doi.org/10.1111/j.1752-1688.2001.tb05519.x.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12 (85), 2825–2830.

Santopietro, S., Gargano, R., Granata, F., de Marinis, G., 2020. Generation of water demand time series through spline curves. J. Water Resour. Plan. Manag 146 (11), 04020080. http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0001282.

Seabold, S., Perktold, J., 2010. Statsmodels: Econometric and statistical modeling with Python. In: Proceedings of the 9th Python in Science Conference. pp. 92–96. http://dx.doi.org/10.25080/Majora-92bf1922-011.

Selle, B., Hannah, M., 2010. A bootstrap approach to assess parameter uncertainty in simple catchment models. Environ. Model. Softw. 25 (8), 919–926.

Serrà, J., Arcos, J.L., 2014. An empirical evaluation of similarity measures for time series classification. Knowl.-Based Syst. 67, 305–314. http://dx.doi.org/10.1016/j.knosys.2014.04.035.

Shani-Zerbib, A., Liberzon, D., 2018. Data set of wind-waves interactions in the Gulf of Eilat (IUI), June 2017. Mendeley Data 2, http://dx.doi.org/10.17632/kgx4559c67.2.

Shani-Zerbib, A., Liberzon, D., Team, T.-S., 2017. Wind-waves interactions in the Gulf of Eilat. In: APS Division of Fluid Dynamics Meeting Abstracts. pp. D19–010.

Shani-Zerbib, A., Rivlin, A., Liberzon, D., 2018. Data set of wind–waves interactions in the Gulf of Aqaba. Int. J. Ocean Coastal Eng 01 (03), 1850003. http://dx.doi.org/10.1142/S2529807018500033.

Srikanthan, R., McMahon, T.A., 2001. Stochastic generation of annual, monthly and daily climate data: A review. Hydrol. Earth Syst. Sci. 5 (4), 653–670. http://dx.doi.org/10.5194/hess-5-653-2001.

Talbot, P.W., Rabiti, C., Alfonsi, A., Krome, C., Kunz, M.R., Epiney, A., Wang, C., Mandelli, D., 2020. Correlated synthetic time series generation for energy system simulations using Fourier and ARMA signal processing. Int. J. Energy Res. 44 (10), 8144–8155. http://dx.doi.org/10.1002/er.5115.

Wiese, M., Knobloch, R., Korn, R., Kretschmer, P., 2020. Quant GANs: Deep generation of financial time series. Quant. Finance 20 (9), 1419–1440. http://dx.doi.org/10.1080/14697688.2020.1730426.

Wilks, D.S., Wilby, R.L., 1999. The weather generation game: A review of stochastic weather models. Progr. Phys. Geogr: Earth Environ 23 (3), 329–357. http://dx.doi.org/10.1177/030913339902300302.

Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K., 2019. Modeling tabular data using conditional GAN. In: Advances in Neural Information Processing Systems. 32, Curran Associates, Inc..

Yoon, J., Jarrett, D., van der Schaar, M., 2019. Time-series generative adversarial networks. In: Advances in Neural Information Processing Systems. 32, Curran Associates, Inc..