Stochastic Environmental Research and Risk Assessment (2020) 34:23–31
https://doi.org/10.1007/s00477-019-01762-3

ORIGINAL PAPER

Nearest neighbor time series bootstrap for generating influent water quality scenarios

William J. Raseman[1] · Balaji Rajagopalan[1] · Joseph R. Kasprzyk[1] · William Kleiber[2]

Published online: 1 January 2020

Abstract

Understanding influent water quality variability is essential for the long-term planning of potable water systems. To quantify variability and generate realistic influent scenarios, we propose a nonparametric time series approach based on k-nearest neighbor (k-NN) bootstrap resampling. The k-NN approach resamples historical data conditioned on a “feature vector” at a given time to generate values at subsequent times. We modified this algorithm by adding random perturbations to the resampled values to generate realistic extremes unobserved in the historical record. k-NN is widely used in stochastic hydrology and hydroclimatology; however, it is adapted here for the multivariate, data-limited context of water treatment. To examine the performance of the algorithm, we applied it to an eleven-year, monthly water quality dataset of alkalinity, temperature, total organic carbon, and pH from the Cache la Poudre River in Colorado. We found that the k-NN simulations captured the relevant distributional statistics of the historical record, which suggests that the algorithm produces realistic and varied scenarios. When used in conjunction with modeling and optimization, these scenarios have the potential to improve the sustainability, resilience, and efficiency of potable water systems.

1 Introduction

Influent water quality conditions largely determine operational decisions at a drinking water treatment plant. These decisions may include altering chemical doses, blending source waters, or backwashing filters. Quantifying the variability and uncertainty of the influent can promote regulatory compliance, public health, and the sustainability of water treatment (Benke and Hamilton 2008; Santana et al. 2014; Towler et al. 2009). The goal of such quantification is to create a suite of realistic water quality scenarios that a treatment plant may experience. To estimate the performance of operating alternatives during such scenarios, several simulation models have been developed (Baxter et al. 1999; Harrington et al. 1992; Maier et al. 2004; Rietveld et al. 2010; Worm et al. 2010). Water managers, consultants, or operators can use simulation to test a range of plausible influent water quality scenarios and estimate which operating policies yield the best performance.

Influent water quality is a function of both natural and anthropogenic impacts, such as seasonal changes, industrial discharge, extreme events (Delpla et al. 2009; Khan et al. 2017), and source water type. Riverine sources tend to have more volatile water quality than lakes and reservoirs, which have the more buffering capacity. Processes that drive water quality within a watershed include erosion and sediment transport, the growth and decay of organic matter, nutrient cycling, and heat transfer. Therefore, quality tends to be a function of the climate and land use characteristics (Delpla and Rodriguez 2014; Samson et al. 2016). Due to this coupled nature of hydrology and water quality, stochastic modeling approaches—which are well developed in hydrology and hydroclimatology (e.g., Khalili et al. 2009; Modarres 2007; Bras and Rodríguez-Iturbe 1985)—are promising for the generation of influent water quality scenarios.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00477-019-01762-3) contains supplementary material, which is available to authorized users.

✉ William J. Raseman
william.raseman@colorado.edu

[1] Department of Civil, Environmental, and Architectural Engineering, University of Colorado Boulder, Boulder, CO 80309, USA

[2] Department of Applied Mathematics, University of Colorado Boulder, Boulder, CO 80309, USA

Stochastic modeling is divided into parametric and nonparametric techniques. Popular parametric techniques include the Thomas-Fiering or autoregressive lag-1 (Maass et al. 1962; Thomas and Fiering 1962) and the autoregressive moving average (Box et al. 2015) models. Nonparametric approaches include kernel density estimation (Sharma et al. 1997), moving block bootstrap (Vogel and Shallcross 1996), and k-Nearest Neighbor (k-NN) bootstrap resampling (Lall and Sharma 1996). Parametric time series models have been used to generate influent data (e.g., Li et al. (2014)) but they require large datasets to produce realistic scenarios—especially, considering multivariate applications. Since such large influent water quality datasets are not often available to water utilities (Towler et al. 2009), nonparametric approaches have promise in this context.

The k-NN bootstrap resampling is perhaps the most commonly used nonparametric approach. It requires few prior assumptions about the data, has shown to be effective for linear and nonlinear relationships, and is able to capture persistence in time series data (Lall and Sharma 1996; Rajagopalan and Lall 1999). The characteristics of streamflow data which motivated the development of the k-NN resampling algorithm—serial dependence, long-term persistence, and nonlinearity (Lall and Sharma 1996)—are also common to influent water quality data. Moreover, Yates et al. (2003) later modified the algorithm to account for intervariable correlations, improving the quality of multivariate simulations. This modification is critical for the viability of these methods for water treatment applications, which are inherently multivariate problems. Sharif and Burn (2007) have further altered the k-NN resampling approach to simulate extreme events beyond those observed in the historical record.

Understanding extremes is critical for long-term decision making and risk management planning (Haimes 2015), and therefore, desirable to simulate for water treatment applications. Extreme weather-related events, such as drought, heat waves, and flooding, can degrade source water quality and disrupt treatment operations (Khan et al. 2017). To mitigate these problems—particularly those related to acute health risks—water utilities may need to issue boil advisories and alter treatment and distribution practices. Due to the short length of water quality datasets, extremes may not be represented within the historical record. Moreover, extreme events are expected to increase in frequency and intensity due to climate change (IPCC 2014). Therefore, it is important for water quality simulations account for these events.

In this work, we adapt the methods described by Lall and Sharma (1996) and modifications by Yates et al. (2003) and Sharif and Burn (2007) to generate realistic scenarios of influent water quality for water treatment applications. To illustrate this modified k-NN bootstrap resampling approach, we simulate from a monthly dataset of Cache la Poudre River water quality. This data exhibits complex interrelationships among water quality variables, serial dependence, and seasonality, which makes it well suited for this technique. To evaluate the performance of the algorithm on the dataset, we compare the sample statistics of the observed record to those of the simulated scenarios. These statistics include the mean, standard deviation, minimum, maximum, joint correlation, and lag-1 autocorrelation. Furthermore, we investigate how simulating extremes impacts algorithmic performance across these metrics.

2 Materials and methods

2.1 Data collection and processing

Water quality data were obtained from the City of Fort Collins Utility for the influent sampling location of the Cache la Poudre River. This eleven-year dataset includes measures of alkalinity, pH, temperature, and total organic carbon (TOC) collected at various frequencies (e.g., 15-min, daily, weekly) by the utility. These water quality variables were chosen due to their impact on the core processes of water treatment: coagulation, filtration, and disinfection. More generally, these variables influence the acid–base chemistry of the water and are key predictors for the formation of contaminants of concern known as disinfection byproducts.

To perform k-NN simulation (see Sect. 2.2), a complete dataset is necessary and each variable must be aggregated to the same frequency. Based on the number of missing data points and timescales relevant for decision-making, we chose a monthly frequency. Before aggregating the data, systematic data collection errors were identified and the associated values were removed from the dataset. This process is detailed in *SI.2 Data Quality Control and Quality Assurance*. After monthly aggregation (n = 132 for each parameter), three data points were missing (n = 129). To create a complete time series dataset (Fig. 1), the *na.interp()* function within the *forecast* package in R was used to interpolate between missing time series values (Hyndman et al. 2018; Hyndman and Khandakar 2007). For seasonal data, *na.interp* uses STL (Seasonal and Trend decomposition using Loess) for this interpolation.

The result of data collection and processing is the observed monthly water quality dataset, $\mathbf{x_t}$, where $\mathbf{x_t}$ denotes the vector of length p of available variables at monthly time point $\mathbf{t}$ = [1,…,12 N], where N is the number of years on record.

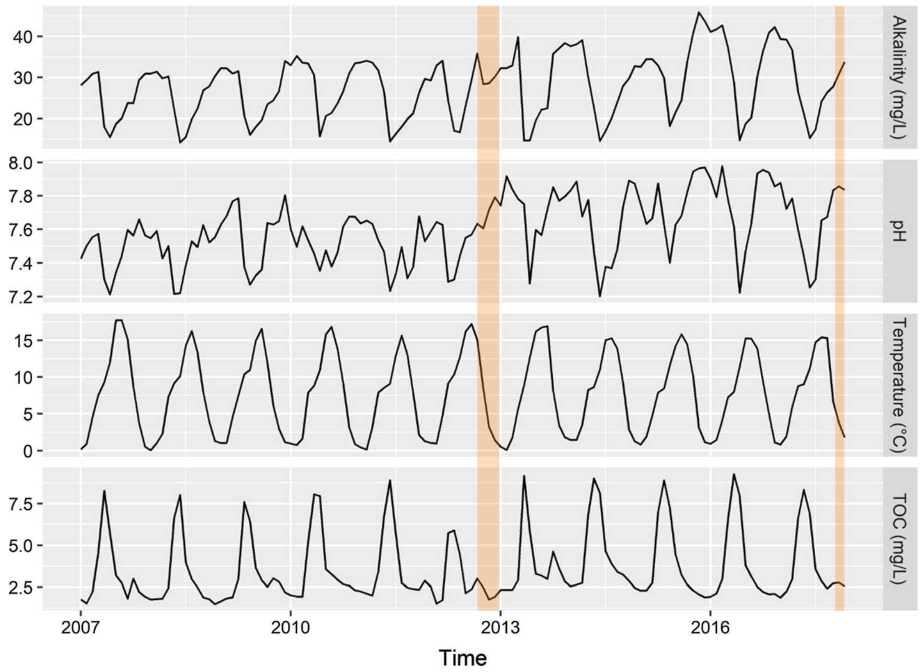

Fig. 1 Monthly Cache la Poudre River influent water quality time series (n = 132) for alkalinity, pH, temperature, and total organic carbon (2007–2017). Regions shaded in orange represent missing values that were filled in using time series interpolation

2.2 Nearest neighbor resampling algorithm

In this section, we describe a six-step, k-Nearest Neighbor (k-NN) bootstrap resampling algorithm—based on the algorithm detailed in Sharif and Burn (2007)—for generating influent water quality scenarios from historical data. To illustrate the algorithm, we apply it to the monthly dataset described above. In this paper, we decided to standardize the simulated data for each month by standardized by subtracting the monthly mean from each observation and dividing by the monthly standard deviation. The algorithm is carried out in standardized space. Then, the simulated data are transformed back to the original space. For information on the implications of standardization, please refer to Section 2.3.

To begin a simulation, the starting month must be defined by the user (e.g., January). The water quality values for the first simulated month are then determined based on randomly selecting a year from the historical record. The water quality that was observed on the user-defined month and randomly selected year serves as the first month of simulated data. The simulation of subsequent months proceeds as follows:

1. *Define a feature vector*: Define a "feature vector", $\boldsymbol{D}_t$, dimension $d = pL$, where L and p are the number of lags and variables considered in the model, respectively. The choice of feature vector represents the dependence on which the simulated values, known as the "successors", are conditioned. For instance, for the case of $L = 1$ and $p = 4$, the generation of successors would be based solely on characteristics of the nearest neighbors of the previous month of the observed data for four different water quality variables: $\boldsymbol{D}_t = \mathbf{x}_{t-1}$. Additional lags can be represented (e.g., $\boldsymbol{D}_t = [\mathbf{x}_{t\text{-}1}, \mathbf{x}_{t\text{-}2}, \ldots, \mathbf{x}_{t-L}]$, where $\boldsymbol{D}_t$ is a 4 × L matrix). To select the appropriate number of lags, the user can use similar approaches to fitting autoregressive and moving average models. These methods include identifying significant lags in the autocorrelation and partial autocorrelation functions of the data and using AIC-based methods (Lee and Ouarda 2011). In our analysis, we identified a lag-1 dependence or $L = 1$ (see *SI.3 Identification of Serial Dependence*).
2. *Find nearest neighbors*: For the current timestep, i, the feature vector, $\boldsymbol{D}_i$, is constructed. The neighbors to the current feature vector, $\boldsymbol{D}_t$, include all years on record for that month. Next, the Mahalanobis distance (Mahalanobis 1936) (Eq. 1) is calculated to determine which neighbors are the nearest to $\boldsymbol{D}_i$ (Sharma and O'Neill 2002; Yates et al. 2003):

$$d_i = \sqrt{(\boldsymbol{D}_t - \boldsymbol{D}_i)^T \boldsymbol{S}^{-1} (\boldsymbol{D}_t - \boldsymbol{D}_i)} \qquad (1)$$

where $\boldsymbol{S}$ is a $p \times p$ matrix which describes the covariance between $\boldsymbol{D}_i$ and $\boldsymbol{D}_t$, thus $\boldsymbol{d}_i$ is an N-dimensional distance vector. N is defined as the number of years on record.
3. *Rank nearest neighbors and select k neighbors*: Once the distance from the feature vector is calculated for each neighbor, the neighbors are ranked in ascending distance order. The nearest k neighbors are then chosen, from which the successor will be selected. Here we use a popular heuristic suggested in Lall and

Sharma (1996) in which $k = \sqrt{N}$. Alternatively, k can be selected using a AIC-based approach described in Lee and Ouarda (2011).

4. *Choose successor*: To probabilistically select neighbors among the k nearest neighbors, $\boldsymbol{D}_t^{kNN}$ ($4 \times k$ matrix, a subset of $\mathbf{D_t}$), we define a weighting function based on the discrete kernel K described in Lall and Sharma (1996). This kernel assigns the greatest probability for the 1st nearest neighbor being selected and the least probability for the kth neighbor. By computing the cumulative distribution function of corresponding to the kernel, we generate a value from uniform distribution on the interval [0,1] to choose which of the k neighbors is selected. For the case of lag-1 dependence ($L = 1$) the selected neighbor, $\boldsymbol{x}_{t-1}$, would produce a successor (i.e., simulated value), $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}$, equal to $\boldsymbol{x}_{\boldsymbol{t}}$.
5. *Add random innovations to successor*: The fifth step of the algorithm has three parts: (a) generate modified successors, (b) bound variables, and (c) check that bounds are met.

   (a) To simulate values beyond those in the observed record, random innovations or errors are added to the successor from Step 4, $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}$. To generate modified successor values, $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{\prime}$, a smoothed bootstrap with variance correction is applied (Eq. 2) (Silverman 1986):

   $$\tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{\prime} = \tilde{\boldsymbol{x}}_{\boldsymbol{i}} + \frac{\boldsymbol{\sigma}_t^{kNN}\boldsymbol{\lambda}_t^{kNN}\boldsymbol{z}_i}{\sqrt{1 + \frac{\boldsymbol{\lambda}_t^{kNN2}\boldsymbol{\sigma}^{K2}}{\boldsymbol{\sigma}_t^{kNN2}}}} \tag{2}$$

   where $\boldsymbol{\sigma}_t^{kNN}$ is the standard deviation of the k nearest neighbors, $\boldsymbol{D}_t^{kNN}$. $\boldsymbol{\lambda}_t^{kNN}$ is the bandwidth of a nonparametric distribution fit to the k nearest neighbors for each variable. Here, we define the bandwidth based on the rule-of-thumb estimation of a Gaussian kernel density estimator—specifically, the *bw.nrd0()* function in the *stats* package in R from Silverman (1986)—due to small sample sizes. z_i is a Gaussian random variate with mean zero and standard deviation of one. $\boldsymbol{\sigma}^{\boldsymbol{K}}$ is the standard deviation of the kernel. In this case, we use a Gaussian kernel, where $\boldsymbol{\sigma}^{\boldsymbol{K}} = 1$**.** Our approach to random innovations differs from the Sharif and Burn algorithm in two ways: 1) it uses a smoothed bootstrap with variance correction and 2) it generates values from the random variate, z, for each variable independently.

   (b) If any water quality variables are bounded—for example, TOC must be non-negative, a variable kernel density estimation method for in Terrell and Scott (1992) may be applied. Specifically, the method can be used to generate a modified bandwidth,$\lambda_t^{kNN'}$, which is less likely to produce negative values than $\boldsymbol{\lambda}_t^{kNN}$. The modified bandwidth is only selected at values near zero, for which the original kernel is likely to simulate negative values. The algorithm we use to accomplish this is the same as that described in Sharma and O'Neill (2002) and Sharif and Burn (2007).

   (c) If bounded, repeat Steps 5a and b until they produce a non-negative value.

6. *The successor determines the next timestep and the process repeats*: Once the modified successor is simulated, $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{\prime}$, it determines the new current timestep. Then, Steps 1–5 are repeated for the following months until the simulated values are equal in length to the historical record. In this paper, we define "one simulation" as a series of simulated values of a length equivalent to the historical record. The user specifies the number of simulations generated by a single run of the algorithm. Those simulations are collectively referred to as an "ensemble".

2.3 Model application

When applying the k-NN resampling algorithm for simulation, the user must choose whether to enable random innovations, whether to standardize the data, which variables must be non-negative, and select the number of simulations. These parameters are each model inputs for the open source code associated with this work and can be modified by the user (see *SI.1 Code and Data Availability*).

The user should enable random innovations if they wish to simulate values beyond the observed record (i.e., extremes). In this work, we chose to standardize the data to avoid the simulation of unrealistically high or low values due to the inclusion of random innovations (discussed in Step 5 of the algorithm). Whether or not standardization is preferred will vary among datasets and requires the user to judge what values they deem to be unrealistic. The magnitude of random innovations can be fine-tuned by adjusting the bandwidth calculation type (e.g., plug-in, least-squares cross validation) in Step 5.

Although random innovations and standardization control the nature of extreme values, it may also be necessary to bound variables that must be non-negative. For instance, a negative value for the concentration of organic carbon would be an unphysical result; therefore, the user must bound this water quality variable to ensure that simulated values are non-negative.

The appropriate number of simulations can be determined empirically by comparing the distributions of simulated data. The goal of this exercise is to determine the minimum number necessary to stabilize the distributional statistics of the simulated data. In other words, if the data from 500 simulations is nearly identical to the data that results from 2,500 and 10,000 simulations, there is likely no benefit to simulate more than 500 water quality scenarios.

In the remainder of the paper, we will examine the performance of our modified k-NN algorithm for the Cache la Poudre dataset. For this dataset, we have enabled random innovations and standardized the simulated data and bounded simulated total organic carbon concentrations to be nonnegative. We have chosen to generate 2500 water quality simulations, each the same length as the historical record. Next, we performed a comparative analysis of monthly statistics for the simulated ensemble and the historical record. The sample statistics that were calculated include the maximum, minimum, mean, standard deviation, lag-1 autocorrelation, and joint correlation. These statistics are commonly used to assess simulation performance in time series literature.

3 Results and discussion

Pairwise boxplots are useful to visually compare the sample statistics between the observed and simulated data. If the simulated data were to reproduce the observed statistics exactly, the extents of the boxes, whiskers, and the median for both datasets would be identical. For all boxplots in this paper, the boxes represent the 25th and 75th percentiles and the whiskers extend no farther than 1.5 times the interquartile range beyond the box. In Fig. 2, we find that the boxplots are similar, overall, suggesting that the distribution is well preserved in the simulated data. Extreme values, represented as points that lie past the whiskers, are produced due to the incorporation of random innovations from Step 5 of the k-NN algorithm. If desired, the random innovations can be disabled, which would remove unobserved values from the simulations, as demonstrated in Fig. 3. Note the difference in axes extents between these two figures.

To rigorously test the similarity of the simulated and observed distributions, we performed a two sample Kolmogorov–Smirnov (K–S) test. This K–S test compares the empirical distribution functions for two samples—the observed and simulated data—with the null hypothesis that the samples come from the same distribution. In our analysis of the monthly distributions for each water quality variable, we found no significant *p* values which suggests that the simulated and observed distributions are similar. This result was true for k-NN simulations both with and without random innovations.

To compare the sample statistics of the observed record and the ensemble of simulated scenarios, we show the boxplots of the monthly mean, minimum, maximum, and standard deviation of TOC (Fig. 4). Although not visualized in Fig. 4, the TOC data is representative of the other water quality variables considered in this work. For the observed record, the sample statistics are visualized as a single point for each month. Ideally, those points should fall close to the median values of boxplots. Thus, Fig. 4

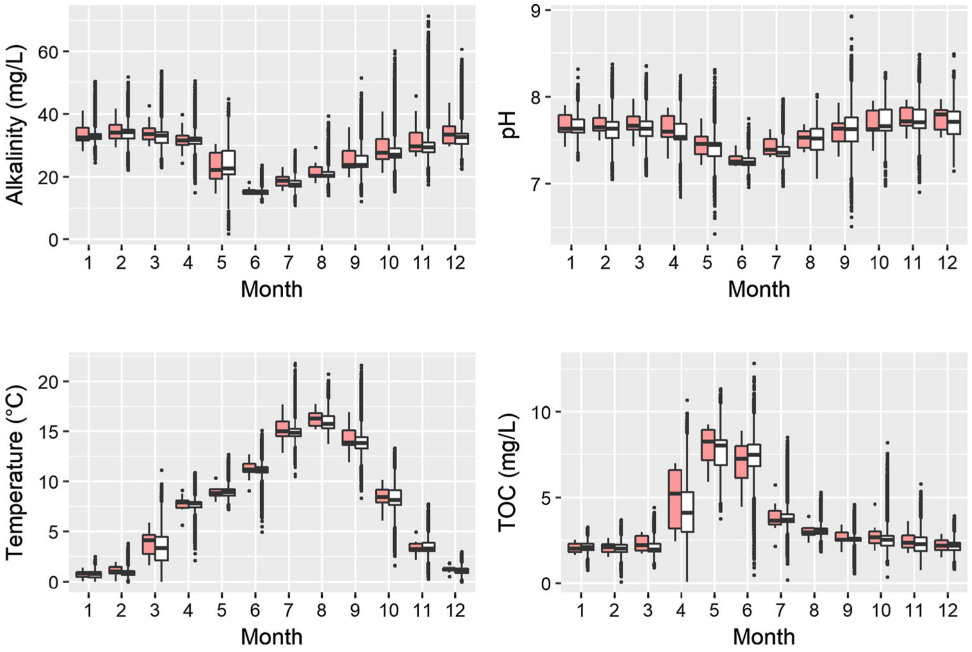

Fig. 2 Boxplots of observed (red boxplots, n = 132) and k-NN simulated (white boxplots, n = 330,000) influent water quality data of the alkalinity, pH, temperature, and total organic carbon for the Cache la Poudre River (2007–2017) in Fort Collins, CO

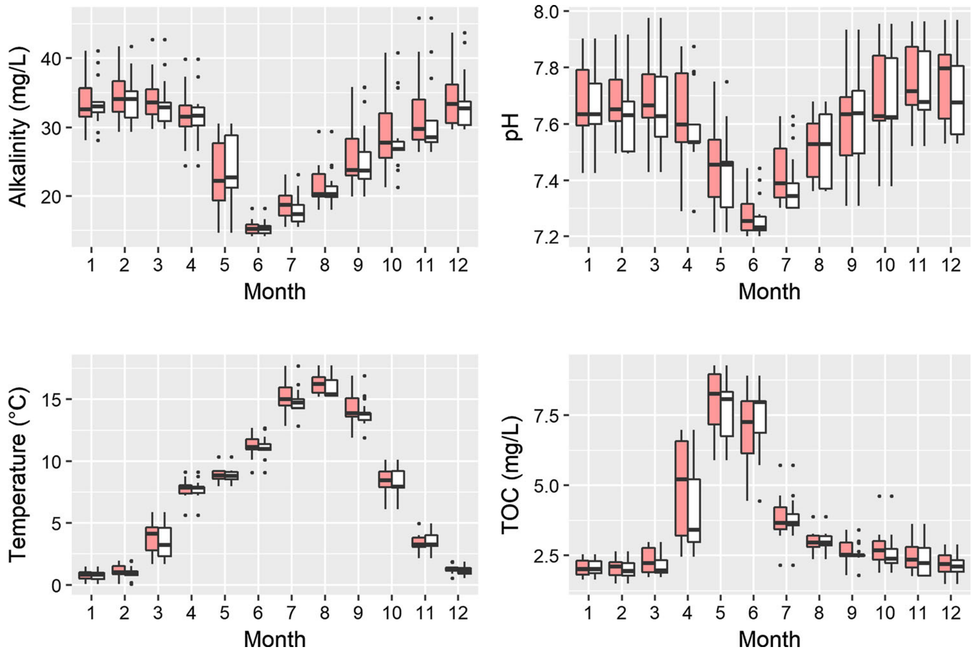

Fig. 3 Boxplots of observed (red, n = 132) and k-NN simulated (white, n = 330,000) influent water quality data of the alkalinity, pH, temperature, and total organic carbon for the Cache la Poudre River (2007–2017) in Fort Collins, CO. In this instance, random innovations have been disabled within the k-NN algorithm

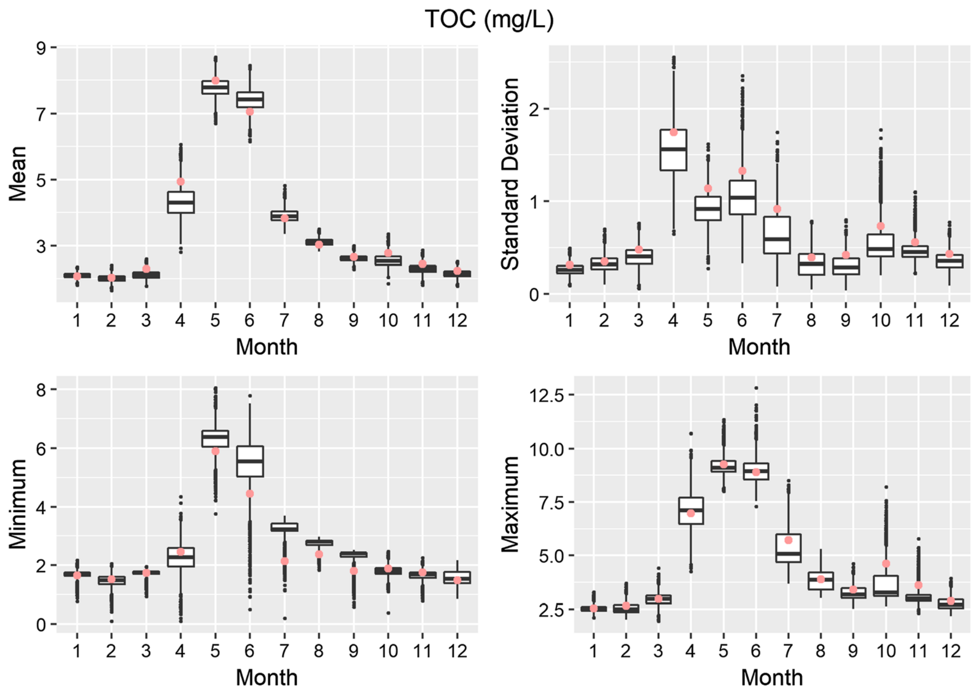

Fig. 4 Observed (red points) and simulated (white boxplots, n = 2500) sample statistics—mean, standard deviation, minimum, and maximum—for total organic carbon data from the Cache la Poudre River (2007–2017) in Fort Collins, CO

shows that these statistics were generally well reproduced by the simulations, with one exception. The minimum and standard deviation tended to be over- and under-simulated, respectively, in the months of April through September.

In addition to preserving the distributional statistics of individual water quality variables, the simulations should also maintain the joint correlation between variables. This is important because joint correlations may be significant to water treatment decision making. For example, alkalinity and pH tend to be highly correlated (see Fig. 1) because they both are related the acid–base reactions. Furthermore, such reactions are significant for treatment efficacy in almost every process within the treatment plant. If the simulated data does not maintain the correlation between alkalinity and pH, it may under- or over-represent risk or the cost associated with these variables for decision makers. Therefore, we assessed the pairwise correlation between all four variables for the observed and simulated data over the period of record (eleven years).

Figure 5 contains comparisons of the observed and simulated correlations with historical values represented as a single point and simulated data (n = 2500) represented by boxplots. Here we see that there is minimal variability among the pairwise correlations and the observed correlations generally lie close to the median simulated values. Thus, with respect to joint correlation, the statistic is reproduced both consistently and accurately in the simulated data.

Lastly, we review the month-to-month or lag-1 autocorrelation of the k-NN simulated data (Fig. 6). Unlike the joint correlation statistic described above, which describes the relationship between two different water quality variables, the autocorrelation compares the same variable across timesteps. Specifically, the *lag-1* autocorrelation is the correlation of data between two adjacent timesteps

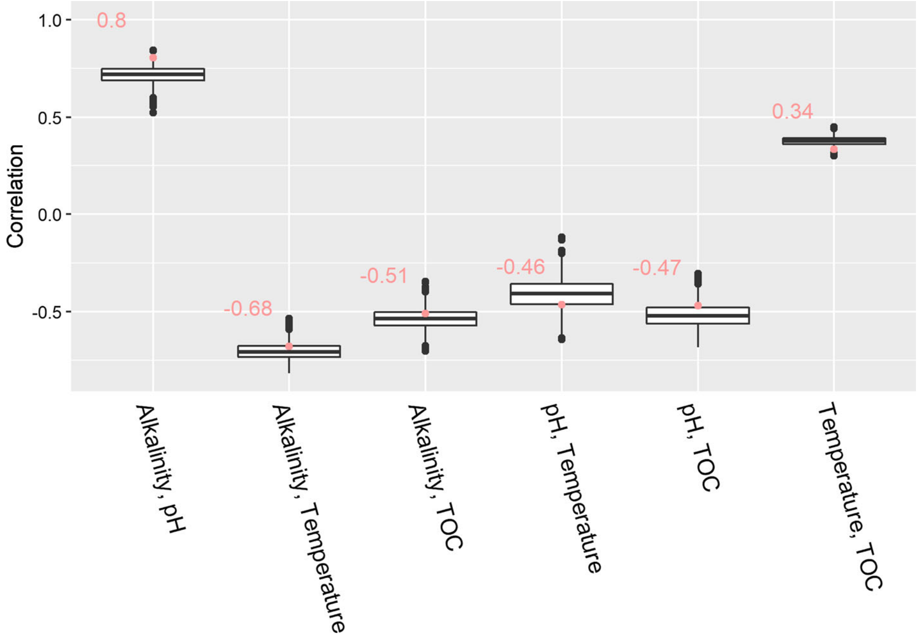

Fig. 5 Observed (red points) and simulated (white boxplots, n = 2500) pairwise correlation statistics for influent alkalinity, pH, temperature, and total organic carbon for the Cache la Poudre dataset. Observed correlations are labeled in red text

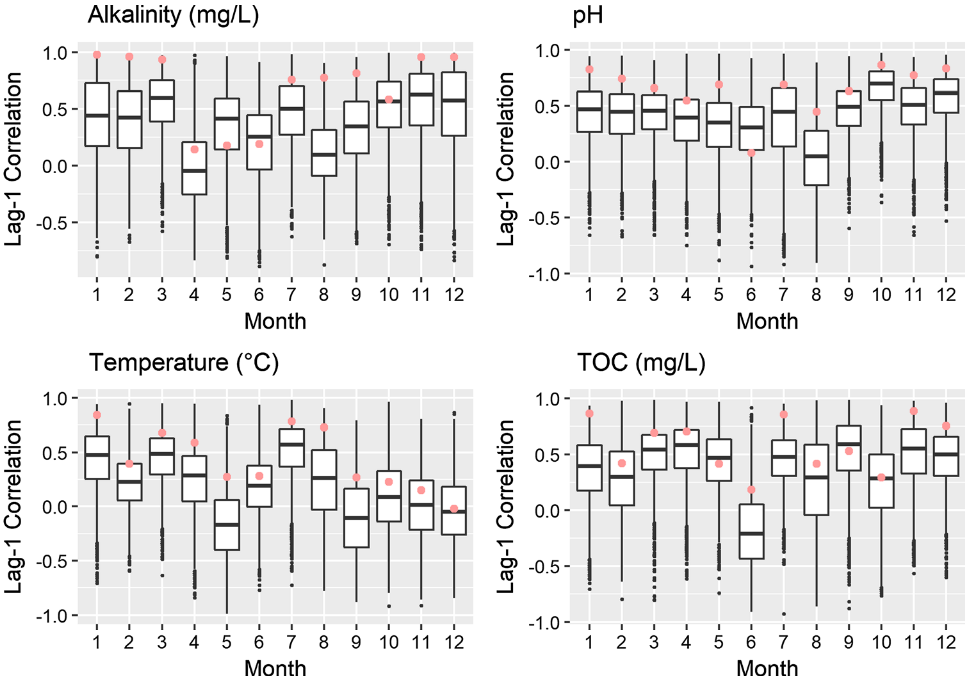

Fig. 6 Observed (red points) and simulated (white boxplots, n = 2500) lag-1 (i.e., month-to-month) autocorrelation for alkalinity, pH, temperature, and total organic carbon data from the Cache la Poudre dataset

(e.g., x_t and x_{t-1}). Our analysis of the autocorrelation function suggests that the first lag is significant in the observed record (see *SI.3 Identification of Serial Dependence*), and therefore, it is a useful statistic to measure of the persistence (i.e., serial dependence) in water quality over time. Comparing the lag-1 autocorrelation of the observed and simulated data (Fig. 6), we find that seasonality of the lag-1 autocorrelation is generally reproduced; however, the autocorrelation tends to be undersimulated across the water quality variables. This is particularly true for alkalinity and pH, which, in most months, have observed autocorrelations near one.

4 Conclusion

Due to the short, multivariate nature of influent water quality datasets, generating influent water quality data which maintain the statistics on the observed record is nontrivial. The k-NN bootstrap resampling algorithm described in this paper is a robust, easy-to-use method for generating influent water quality scenarios. Specifically, we illustrate this approach for monthly data; however, it could be adapted to other time scales of interest. Based on a multivariate water quality dataset of eleven years, we demonstrated that the k-NN algorithm can reproduce the monthly mean, minimum, maximum, standard deviation, lag-1 autocorrelation, and joint correlation for the period of record. Furthermore, by implementing random innovations described by Sharif and Burn (2007), the algorithm can produce values beyond those contained in the observed record.

The synthetic values produced by random innovations allow for the simulation of extreme influent conditions which are significant for long-term planning and risk management. Although it is difficult to quantify whether the synthetic values are realistic, expert opinion can be used to determine their realism on a case-by-case basis. If the variability introduced by the random innovations is found to be too large or too small, the user can adjust settings for data standardization and the bandwidth calculation—discussed in Step 5 of the algorithm. Additionally, for variables that are subject to some upper or lower bound, these variables users can impose these bounds to ensure physically meaningful data.

The simulated data produced by the k-NN algorithm has many potential applications within water treatment and beyond. As discussed in Towler et al. (2009) water quality scenario generation techniques can illustrate uncertainty and variability in influent conditions which provides insight for treatment decision making. This is especially true for long-term planning, in which regulations, demand, source water availability, and influent water quality are subject to change over time (Brookes et al. 2014). From computational perspective, influent water quality scenarios—whether observed or synthetic—are necessary to simulate the impacts of influent conditions.

These scenarios can also be used conjunction with water quality simulation and optimization algorithms (i.e., simulation–optimization). Simulation–optimization methods can suggest best practices for operational and infrastructural improvements for water treatment (Raseman et al. 2017). Similar techniques are used in water resources and water quality management in which synthetic time series data are fed into simulation–optimization schemes to generate sets of optimal solutions (Quinn et al. 2017; Ward et al. 2015). Furthermore, this technique is not exclusive to water treatment applications. Similar approaches have been outlined in hydrology (Lall and Sharma 1996; Sharma and O'Neill 2002) and hydroclimatology (Rajagopalan and Lall 1999; Sharif and Burn 2007; Yates et al. 2003) and these methods could also be applied to related fields such as wastewater treatment.

Lastly, as real-time water quality monitoring continues to advance, dataset quality will improve which will improve the performance of the k-NN resampling algorithm. Moreover, this algorithm will become increasingly relevant as both mechanistic and data-driven treatment simulators grow in popularity.

Acknowledgements The authors would like to thank Jill Oropeza and Jared Health at City of Fort Collins Utilities for providing the Cache la Poudre water quality data. This work was supported by the U.S. Environmental Protection Agency "National Priorities: Systems-Based Strategies to Improve the Nation's Ability to Plan and Respond to Water Scarcity and Drought Due to Climate Change", Grant No. R835865. The contents of this manuscript are solely the responsibility of the grantee and do not necessarily represent the official views of the US EPA.

References

Baxter CW, Stanley SJ, Zhang Q (1999) Development of a full-scale artificial neural network model for the removal of natural organic matter by enhanced coagulation. J Water Serv Res Technol Aqua 48:129–136

Benke KK, Hamilton AJ (2008) Quantitative microbial risk assessment: uncertainty and measures of central tendency for skewed distributions. Stoch Environ Res Risk Assess 22:533–539. https://doi.org/10.1007/s00477-007-0171-9

Box GE, Jenkins GM, Reinsel GC, Ljung GM (2015) Time series analysis: forecasting and control. Wiley, Hoboken

Bras RL, Rodríguez-Iturbe I (1985) Random Functions and Hydrology. Courier Corporation, North Chelmsford

Brookes JD, Carey CC, Hamilton DP, Ho L, van der Linden L, Renner R, Rigosi A (2014) Emerging challenges for the drinking water industry. Environ Sci Technol 48:2099–2101. https://doi.org/10.1021/es405606t

Delpla I, Rodriguez MJ (2014) Effects of future climate and land use scenarios on riverine source water quality. Sci Total Environ 493:1014–1024

Delpla I, Jung A-V, Baures E, Clement M, Thomas O (2009) Impacts of climate change on surface water quality in relation to drinking water production. Environ Int 35:1225–1233. https://doi.org/10.1016/j.envint.2009.07.001

Haimes YY (2015) Risk modeling, assessment, and management. Wiley, Hoboken

Harrington GW, Chowdhury ZK, Owen DM (1992) Developing a computer model to simulate dbp formation during water treatment. J Am Water Works Assoc 84:78–87

Hyndman RJ, Khandakar Y (2007) Automatic time series for forecasting: the forecast package for R. Monash University, Department of Econometrics and Business Statistics.

Hyndman RJ, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, O'Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F (2018) forecast: forecasting functions for time series and linear models. R package.

IPCC (2014) Climate Change 2014: synthesis report. Contribution of working groups I, II and III to the fifth assessment report of the intergovernmental panel on climate change.

Khalili M, Brissette F, Leconte R (2009) Stochastic multi-site generation of daily weather data. Stoch Environ Res Risk Assess 23:837–849. https://doi.org/10.1007/s00477-008-0275-x

Khan SJ, Deere D, Leusch FD, Humpage A, Jenkins M, Cunliffe D, Fitzgerald SK, Stanford BD (2017) Lessons and guidance for the management of safe drinking water during extreme weather events. Environ Sci Water Res Technol 3(2):262–77. https://doi.org/10.1039/C6EW00165C

Lall U, Sharma A (1996) A nearest neighbor bootstrap for resampling hydrologic time series. Water Resour Res 32:679–693. https://doi.org/10.1029/95WR02966

Lee T, Ouarda TBMJ (2011) Identification of model order and number of neighbors for k-nearest neighbor resampling. J Hydrol 404:136–145. https://doi.org/10.1016/j.jhydrol.2011.04.024

Li Z, Clark RM, Buchberger SG, Jeffrey Yang Y (2014) Evaluation of climate change impact on drinking water treatment plant operation. J Environ Eng 140:A4014005. https://doi.org/10.1061/(ASCE)EE.1943-7870.0000824

Maass A, Hufschmidt MM, Dorfman R, Thomas HA, Marglin SA, Fair GM, Bower BT, Reedy WW, Manzer DF, Barnett MP (1962) Design of water resource systems; new techniques for relating economic objectives, engineering analysis and governmental planning. Harvard University Press

Mahalanobis PC (1936) On the generalized distance in statistics. National Institute of Science of India

Maier HR, Morgan N, Chow CWK (2004) Use of artificial neural networks for predicting optimal alum doses and treated water quality parameters. Environ Model Softw 19:485–494. https://doi.org/10.1016/S1364-8152(03)00163-4

Modarres R (2007) Streamflow drought time series forecasting. Stoch Environ Res Ris Assess 21:223–233. https://doi.org/10.1007/s00477-006-0058-1

Quinn JD, Reed PM, Giuliani M, Castelletti A (2017) Rival framings: a framework for discovering how problem formulation uncertainties shape risk management trade-offs in water resources systems. Water Resour Res 53:7208–7233. https://doi.org/10.1002/2017WR020524

Rajagopalan B, Lall U (1999) A k-nearest-neighbor simulator for daily precipitation and other weather variables. Water Resour Res 35:3089–3101

Raseman WJ, Kasprzyk JR, Rosario-Ortiz FL, Stewart JR, Livneh B (2017) Emerging investigators series: a critical review of decision support systems for water treatment: making the case for incorporating climate change and climate extremes. Environ Sci Water Res Technol 3:18–36. https://doi.org/10.1039/C6EW00121A

Rietveld LC, van der Helm AWC, van Schagen KM, van der Aa LTJ (2010) Good modelling practice in drinking water treatment, applied to Weesperkarspel plant of Waternet. Environ Model Softw Thematic Issue Model Autom Water Wastewater Treat Process 25:661–669. https://doi.org/10.1016/j.envsoft.2009.05.015

Samson CC, Rajagopalan B, Summers RS (2016) Modeling Source Water TOC Using Hydroclimate Variables and Local Polynomial Regression. Environ Sci Technol 50:4413–4421. https://doi.org/10.1021/acs.est.6b00639

Santana MVE, Zhang Q, Mihelcic JR (2014) Influence of water quality on the embodied energy of drinking water treatment. Environ Sci Technol 48:3084–3091. https://doi.org/10.1021/es404300y

Sharif M, Burn D (2007) Improved K -nearest neighbor weather generating model. J Hydrol Eng 12:42–51. https://doi.org/10.1061/(ASCE)1084-0699(2007)12:1(42)

Sharma A, O'Neill R (2002) A nonparametric approach for representing interannual dependence in monthly streamflow sequences. Water Resourc Res 38(7):1100. https://doi.org/10.1029/2001WR000953

Sharma A, Tarboton DG, Lall U (1997) Streamflow simulation: a nonparametric approach. Water Resour Res 33:291–308

Silverman BW (1986) Density estimation for statistics and data analysis. CRC Press, Boca Raton

Terrell GR, Scott DW (1992) Variable kernel density estimation. Ann Stat 20:1236–1265

Thomas, H.A., Fiering, M.B., 1962. Mathematical synthesis of streamflow sequences for the analysis of river basin by simulation. In: Design of water resources-systems pp 459–493.

Towler E, Rajagopalan B, Seidel C, Summers RS (2009) Simulating ensembles of source water quality using a k-nearest neighbor resampling approach. Environ Sci Technol 43:1407–1411. https://doi.org/10.1021/es8021182

Vogel RM, Shallcross AL (1996) The moving blocks bootstrap versus parametric time series models. Water Resour Res 32(6):1875–82

Ward VL, Singh R, Reed PM, Keller K (2015) Confronting tipping points: Can multi-objective evolutionary algorithms discover pollution control tradeoffs given environmental thresholds? Environ Model Softw 73:27–43. https://doi.org/10.1016/j.envsoft.2015.07.020

Worm GIM, van der Helm AWC, Lapikas T, van Schagen KM, Rietveld LC (2010) Integration of models, data management, interfaces and training support in a drinking water treatment plant simulator. Environ Model Softw Thematic Issue Model Autom Water Wastewater Treat Process 25:677–683. https://doi.org/10.1016/j.envsoft.2009.05.011

Yates D, Gangopadhyay S, Rajagopalan B, Strzepek K (2003) A technique for generating regional climate scenarios using a nearest-neighbor algorithm. Water Resour Res. https://doi.org/10.1029/2002WR001769

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