Machine Learning–Based Source Identification in Sewer Networks

Aly K. Salem, S.M.ASCE[1]; and Ahmed A. Abokifa, Ph.D., A.M.ASCE[2]

Abstract: Motivated by the valuable epidemiological information it reveals, wastewater surveillance has received significant attention in recent years. Furthermore, monitoring the water quality in sewer systems has been shown to provide useful information to support wastewater treatment operations. Yet, a critical need still exists for developing novel approaches for rapid and efficient source identification of chemical and biological species of interest in sewer systems. A limited number of source identification approaches have been proposed in previous literature, and the majority of these approaches employed various simplifying assumptions that limit their usage in real-life applications. In this study, a machine learning–based simulation-optimization framework was developed to determine the characteristics (i.e., concentration and loading pattern) of multiple simultaneous injection sources in sewer systems. The simulation was conducted using a surrogate model in the form of a multilayer perceptron neural network, which was trained using simulation results derived from the Storm Water Management Model (SWMM). The simulation model was then coupled with a genetic algorithm to reveal the characteristics of multiple sources that reproduce the concentration patterns observed at one or more monitoring locations in the sewer system. The proposed framework was applied to a range of injection scenarios and was able to identify the characteristics of multiple simultaneous injection sources under different conditions. The results showed that the residence time plays a significant role in the identifiability of the injection source location. The proposed framework is applicable to a wide number of source identification applications, including contamination source identification and wastewater-based epidemiology. **DOI: 10.1061/JWRMD5.WRENG-6050.** *© 2023 American Society of Civil Engineers.*

Author keywords: Source identification; Machine learning; Sewer networks; Contamination detection; Optimization.

Introduction

Both combined and separate sewer networks are vulnerable to contamination, whether intentionally or unintentionally. Contamination in combined sewer networks usually takes place due to the wash-up of contaminants from different land uses after a rain event (Gromaire et al. 2001). In separate sewer networks, contamination can take place due to an intrusion of chemical and/or biological species into one of the junctions discharging to the network, such as underground tanks, gas stations, and parking lots, or through overflows and illicit connections (Panasiuk et al. 2015). Such contamination events could have a significant impact on the performance of wastewater treatment plants (WWTPs) and the quality of the final recipient water body (Diaz-Fierros et al. 2002). For instance, the price of wastewater services in some countries takes into account the quality of the wastewater (Banik et al. 2017). Furthermore, the US and European Union require that operators get a permit before discharging to the sewer networks to protect sewer systems from the intrusion of illicit contaminants (Banik et al. 2017). Thus, there is a crucial need for reliable approaches to identify the sources of various constituents in sewer systems to help in the detection and elimination of contaminants from the network.

In addition to contamination detection, source identification (SI) in sewer systems can provide valuable information to support wastewater-based epidemiology (WBE), which has gained significant attention in recent years (Sangkham 2021), especially in the wake of the COVID-19 pandemic. Several studies confirmed the presence of viral SARS-CoV-2 RNA in the stool samples of patients (Dhama et al. 2021). This triggered a widescale effort to collect wastewater samples from WWTPs around the world (Ahmad et al. 2021), which were shown to provide valuable information for tracking infection trends (Suthar et al. 2021), and thus may act as an early warning for new outbreaks (Zhu et al. 2021). In addition to tracking viral concentrations in wastewater, several countries have been utilizing wastewater sampling to quantitatively measure the trends of illicit drug consumption (Yuan et al. 2020).

To support the aforementioned contamination response and WBE applications, novel source identification algorithms need to be developed. The objective of SI is to determine the characteristics of the source junction(s) of a certain species of interest (e.g., a contaminant or a biomarker) that is discharged into the system. The input to the SI problem is a set of time series concentrations of the constituent observed at one or more sampling locations (e.g., online sensors or grab samples). The accuracy and reliability of the measured concentrations are key to accurate source identification. For contamination events, SI constitutes an integral component of contamination response strategies that start after the detection of a contaminant trace or any other species of interest at a sampling location (e.g., WWTP), and is typically followed by a plan to remove the contaminant from the system. For wastewater-based epidemiology, SI can help identify hot spots of viral infections and quickly reveal the locations of new outbreaks within the sewer shed.

[1] Ph.D. Student, Dept. of Civil, Materials, and Environmental Engineering, Univ. of Illinois Chicago, Chicago, IL 60607; Assistant Lecturer, Faculty of Engineering, Cairo Univ., Giza 12613, Egypt. ORCID: https://orcid.org/0000-0002-2295-9971

[2] Assistant Professor, Dept. of Civil, Materials, and Environmental Engineering, Univ. of Illinois Chicago, Chicago, IL 60607 (corresponding author). ORCID: https://orcid.org/0000-0002-2474-6670. Email: abokifa@uic.edu

Note. This manuscript was submitted on November 15, 2022; approved on March 14, 2023; published online on May 27, 2023. Discussion period open until October 27, 2023; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Water Resources Planning and Management***, © ASCE, ISSN 0733-9496.

In drinking water distribution systems (WDSs), SI has been a concern for a long time for the sake of protecting public health against drinking water contamination (Di Cristo and Leopardi 2008; Yang et al. 2009). As a result, it has been extensively studied by the WDS analysis research community, and different approaches were developed and investigated to identify contamination sources in WDSs (Adedoja et al. 2018). Simulation-optimization approaches, in which a species transport simulation model is coupled with an optimization algorithm, were commonly adopted in the majority of these studies. For instance, Liu et al. (2011) and Preis and Ostfeld (2007) coupled EPANET (Rossman et al. 2000) with an optimization technique to solve the SI problem. Various probabilistic approaches were also proposed in previous literature. For example, Perelman and Ostfeld (2013), Wang and Zhou (2017), and Wang and Harrison (2014) implemented Bayesian methods to find the most probable solution for the SI problem. Additionally, parametric uncertainty and its effect on the SI problem solution were investigated in other studies (Preis and Ostfeld 2011; Vankayala et al. 2009). In addition to WDSs, the SI problem has also been extensively investigated in other water systems, including groundwater (Li and Mao 2011; Li et al. 2021; Sun et al. 2006) and river networks (Lee et al. 2018; Lugão et al. 2022; Wu et al. 2020), with the aim of protecting water resources and supporting environmental remediation.

Despite extensive efforts to develop SI methods for various water systems, the SI problem has not been well studied in the context of sewer networks, with only a handful of studies that aimed to define the problem and propose approaches for its solution. In these studies, the Storm Water Management Model (SWMM) (Rossman 2015) was typically used for performing hydraulic and transport simulations. Banik et al. (2014) developed an ad hoc toolkit that was integrated with SWMM to automate the simulation process. In this study, the SI problem was formulated as an optimization problem and solved using the genetic algorithm (GA). In a later study, a prescreening procedure was implemented to reduce the computational burden of the simulation-optimization framework (Banik et al. 2015). This was done using a pollution matrix concept that was first introduced by Kessler et al. (1998) for WDSs to reduce the number of candidate junctions. Later, Banik et al. (2017) implemented the same approach to identify illicit intrusion in sewer networks under dry and wet weather conditions. They performed sensitivity analysis on the GA parameters, flow variabilities, and sensor measurement errors. The results showed that the model was able to identify the source characteristics in both conditions, and it also showed that it is highly dependent on the input data quality.

In all of the aforementioned studies, several assumptions were made to simplify the SI problem. First, the constituent of interest was assumed to be nonreactive. This was mainly done to avoid the confusion resulting from the combined effects of the residence time and reaction rate, which increases the uncertainty and complexity of the SI problem (Sambito and Freni 2021). Nevertheless, various chemical and biological species (e.g., viral RNA or illicit drugs) are known to undergo decay reactions in sewer systems (Sambito et al. 2020), and thus ignoring such kinetics severely limits the applicability of the SI frameworks developed in previous studies. More importantly, previous studies generally assumed the presence of only a single source in the sewer network. However, in practical applications, the exact number of sources is typically unknown, and in many cases multiple sources might simultaneously exist. Additionally, considering more than one source complicates the problem because the spatial variation of the source locations relative to the observation locations will considerably affect the results.

Another significant limitation in existing SI simulation-optimization methods is the relatively high computational cost needed for the optimization to converge. To reduce the computational cost of SI in WDSs, previous studies have proposed using surrogate data-driven models in place of numerical models. In general, surrogate models are built by utilizing a machine learning technique (Hou et al. 2021), and they have been shown to provide a more computationally efficient alternative to physically based models (Majumder and Lu 2021). As a replacement to EPANET, Broad et al. (2005) implemented artificial neural networks for WDSs optimization, and Preis and Ostfeld (2006) employed hybrid model trees to reduce the computational burden of the SI problem in WDSs. Additionally, Lee et al. (2018) proposed replacing SWMM with a random forest model to identify contamination source locations in river networks. Nevertheless, the implementation of surrogate models for water quality simulation of sewer networks has not been attempted yet.

To address the aforementioned gaps in current knowledge, the objectives of this study are to (1) develop a novel SWMM-based simulation-optimization framework for SI in sewer networks that is capable of revealing the characteristics of multiple sources of a reactive species, (2) create a surrogate model to efficiently conduct constituent transport simulations and implement it within the SI framework, and (3) conduct a comprehensive sensitivity analysis of the accuracy of the proposed framework under different conditions.

Methods

The proposed framework aims to reveal the location of each injection source, along with the injection concentration, start time, and duration. Herein, the SI problem, which is an inverse modeling problem by definition, is solved through a forward simulation-optimization technique that has been shown to produce better accuracy compared to other SI techniques (Hu et al. 2015). In the simulation component, two different approaches were implemented, namely, a numerical model and a machine learning–based model. In the first approach, we linked SWMM, as implemented within the Python-based library PySWMM (McDonnell et al. 2020), directly in the optimization model. Alternatively, in the second approach, a multilayer perceptron neural network (MLP-NN) model was built using simulation results generated by PySWMM for different injection events. The optimization module in both approaches was based on a GA as implemented within the Python-based library PyGAD (Gad 2021). Fig. 1 depicts the workflow of the two approaches within the simulation-optimization technique, which are explained in more detail in the following sections.

Optimization Model Formulation

The objective function was formulated as a minimization problem that aims to reduce the normalized root-mean-square error (nRMSE) between the observed and the simulated concentrations at one or more observation junction(s). The optimal solution to this optimization problem is an injection event with a concentration time series

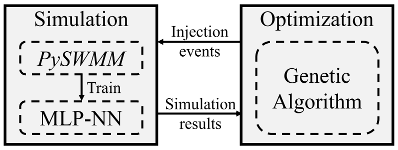

Fig. 1. Proposed SI framework.

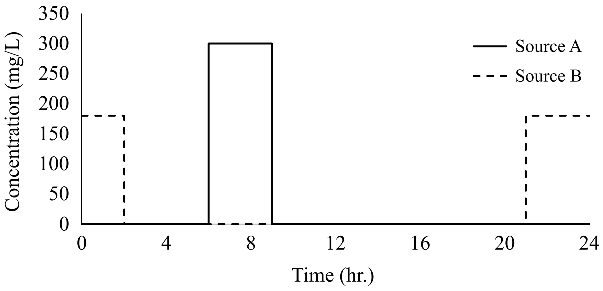

Fig. 2. Pulse-shaped injection pattern in the SI framework for two injection sources: injection at Source A ends within the simulation period, and injection at Source B after the end of the simulation period.

that matches the pattern observed at the monitoring location(s). The objective function of the SI framework is mathematically represented as

$$\text{Minimize nRMSE} = \sum_{j=1}^{m} \frac{\left[\sum_{i=1}^{n} (C_{ji}^{obs} - C_{ji}^{sim})^2 / n\right]^{1/2}}{(\sum_{i=1}^{n} C_{ji}^{obs})/n} \tag{1}$$

where C^{obs} = species concentration observed at the observation junction; C^{sim} = species concentration simulated by the simulation model (either SWMM or MLP-NN); j = index of the observation junction; m = total number of observation junctions; i = time step index; and n = total number of time steps.

In this study, it was assumed that the injection pattern at each junction is a pulse-shaped pattern, where the injection concentration is constant and occurs once during the simulation period (Fig. 2). Hence, for each injection source, the optimization variables are the injection concentration (C) the injection start time (H), and the injection duration (P). A 24-h periodic cycle is implemented for the injection patterns (e.g., the dotted line in Fig. 2). In addition, it was assumed that the real-time sensors located at the observation junctions are capable of measuring the species concentration over time (Kim et al. 2013), and that the decay rate of that species is known.

Unlike previous studies, in which only a single injection source was considered, the proposed SI framework considers all the network junctions to simultaneously act as injection sources. Hence, the number of optimization variables is equal to $3n_J$, where n_J is the number of junctions in the network. The proposed framework also allows for considering a specific number of injection sources ($n_i < n_J$) to reduce the search domain of the SI problem. This is because the complexity of the SI problem, represented by the number of optimization variables, increases significantly with the network size if all the network junctions are considered to be simultaneous sources (Fig. 3). Alternatively, if the number of injection junctions is known, the developed framework expands the set of optimization variables to include the locations of the injection sources. For this case, the number of optimization variables reduces from $3n_J$ to $4n_i$.

In addition to the number of injection sources (n_i), the complexity of the SI problem also depends on the search range of the injection characteristics. Because a 24-h periodic cycle was considered in this study, the range of the injection time and duration was set to be 24 h. Accordingly, the SI problem complexity will be dependent on the number of injection sources (n_i) and the range of injection concentrations (C_{range}). The number of possible alternatives representing the complexity of the SI problem follows Eq. (2). Furthermore, Fig. 3 shows the number of alternatives of a 10-junction network for different values of n_C and C_{range}. This figure shows that the number of alternatives increases significantly by accounting for more injection sources and concentration ranges

$$\text{number of alternatives} = n_J(n_J - 1)^{n_i - 1} C_{\text{range}}^{n_i} 24^{2n_i} \tag{2}$$

Prescreening of Source Junctions

The basis of the SI problem is a time series concentration that is observed at one or more monitoring junction(s). Thus, based on the location of a monitoring junction within the network, the junction may or may not be connected to potential source locations depending on whether the monitoring junction is located downstream (i.e., receives flow) from candidate source locations. Thus, identifying the connectivity status of the network junctions is crucial to building an efficient and responsive surrogate model. Herein, this is done by eliminating the junctions that are not connected to any of the observation junctions from the list of possible injection locations. Thus, the total number of junctions (n_J) used hereinafter denotes only the connected junctions. This prescreening process is particularly important in generating the data sets required to train and test the MLP-NN surrogate model to eliminate redundancies and reduce the computational burden of building the surrogate model and solving the SI problem.

Genetic Algorithm Optimization

The objective function is minimized by the GA, which is a heuristic optimization algorithm that mimics the mechanisms of natural selection and population genetics to produce powerful individuals from weaker parents (Elbeltagi et al. 2005). GA was selected as

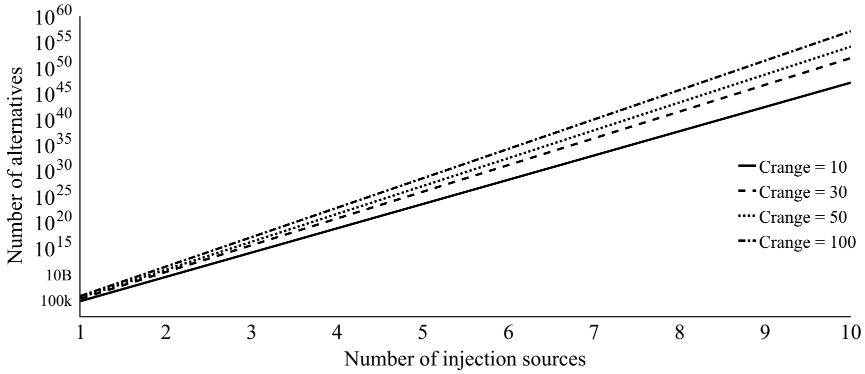

Fig. 3. Relation between the number of possible alternatives in a 10-junction network and the number of injection sources for different ranges of injection concentration.

Table 1. Genetic algorithm optimization parameters

Optimization parameter	Value
Number of individuals	100,000
Parents percentage	1%
Crossover probability	0.9
Mutation probability	0.1
Total generations	40
Unchanged generations	15

the optimization technique due to its superiority in dealing with discontinuous and highly nonlinear functions (Haupt and Haupt 2003). Because the focus of this study is developing a framework for identifying multiple injection sources, we opted for using an optimization algorithm that has been well established in the water and sewer network optimization literature (Pan and Kao 2009; Preis and Ostfeld 2008; Xuesong et al. 2017).

GA starts by randomly generating a set of injection events to form an initial population. These events are then simulated by the MLP-NN surrogate model to evaluate the objective function of each event. Later, GA selects the elite events (i.e., parents) and applies three different operators, namely, selection, crossover, and mutation, to produce a new set of events (i.e., a new population). GA then repeats this process until one of the stoppage criteria listed in Table 1 is reached, and then the fittest event is reported. To employ the GA algorithm in the proposed SI framework, the PyGAD package introduced by Gad (2021) was used, with the optimization parameters shown in Table 1.

Number of Optimization Runs

In the proposed SI framework, each test scenario is performed multiple times to overcome the randomness introduced by the population generation, mutation, and crossover operators of the GA. Also, having different SI solutions for the same test scenario allows for analyzing the accuracy and the precision of the proposed SI framework as described in the "Results and Discussion" section. To determine the sufficient number of optimization runs, an analysis was carried out to observe the change in the objective function (nRMSE) and the results are presented in Section S1. The analysis showed that after 30 optimization runs, the change in the average nRMSE falls below 1%. Hence, a number of 50 optimization runs was conservatively selected in this study.

Forward Simulation Methods

Numerical Model

Hydraulic and transport simulations were performed using SWMM (Rossman 2015). Because the simulation-optimization approach typically involves a large number of simulations, SWMM simulations were conducted using the PySWMM package (McDonnell et al. 2020), which is a Python interface for SWMM developed within the OpenWaterAnalytics (OWA) initiative. PySWMM provides several functions that allow automating the process of setting network and simulation parameters and the extraction of the simulation results in an efficient way.

Surrogate Model

Despite their wide implementation, the key drawback of physically based models (e.g., SWMM) is the relatively large simulation time needed for each run. Thus, running hundreds of thousands of simulations, as is typically needed for optimization, requires a considerably long time. To overcome this limitation, an MLP-NN surrogate model was developed and implemented in place of SWMM. MLP-NN is a supervised machine learning algorithm, capable of learning the behavior of a function by training on an input-output data set. Given the characteristics of a set of injection events as an input, and the corresponding SWMM simulation results as an output, MLP-NN can learn the input-output relationship. This was done by adjusting weighting factors and biases between each layer iteratively through a back-propagation process, giving the MLP-NN the ability to represent highly nonlinear relationships. A comparison between the results of the MLP-NN surrogate model and the SWMM model, together with the computational cost reduction achieved by the MLP-NN surrogate model is demonstrated in the "Results and Discussion" section.

Data Set Generation

Training and testing the MLP-NN surrogate model require several injection events to be generated and simulated by SWMM, where for each injection event, three parameters are randomly generated for each junction in the network. The injection concentration (C) ranges from zero to $C_{\max}$, where zero means no injection and $C_{\max}$ is a user-defined value. The injection start time (H) and the injection duration (P) both range between 1 and 24. The injection parameters of each junction were stacked to form the vector representing each injection event, in the form of $[(C_1, H_1, P_1), (C_2, H_2, P_2), \ldots, (C_{n_J}, H_{n_J}, P_{n_J})]$. The previous process was repeated to produce n_{sim} simulation events, forming an input matrix with the size of $n_{sim} \times 3n_J$. In the process of surrogate model creation, an injection was considered to occur in all network junctions; hence, the locations of the injection sources were not included in the training data set.

Modified SWMM input files were then generated, in which information from each injection event was added to the original input file containing other essential parameters (e.g., layout, elevations, and dimensions) for the network under consideration. These input files were then passed to PySWMM, and the simulation results of each event were extracted at one or more observation location(s). For each observation location j, a concentration time series of size n was obtained from the simulation. The matrix of the simulation output is hence given by

$$\boldsymbol{O}(\boldsymbol{S}) = \begin{bmatrix} \boldsymbol{O}_1(\boldsymbol{S}_1)_1 & \cdots & \boldsymbol{O}_1(\boldsymbol{S}_n)_1 & \cdots & \cdots & \boldsymbol{O}_m(\boldsymbol{S}_1)_1 & \cdots & \boldsymbol{O}_m(\boldsymbol{S}_n)_1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{O}_1(\boldsymbol{S}_1)_{\boldsymbol{n_{sim}}} & \cdots & \boldsymbol{O}_1(\boldsymbol{S}_n)_{\boldsymbol{n_{sim}}} & \cdots & \cdots & \boldsymbol{O}_m(\boldsymbol{S}_1)_{\boldsymbol{n_{sim}}} & \cdots & \boldsymbol{O}_m(\boldsymbol{S}_n)_{\boldsymbol{n_{sim}}} \end{bmatrix} \quad (3)$$

where $\boldsymbol{O}_j(\boldsymbol{S}_i)_e$ = simulation result at time i at the observation junction j for the eth injection event, resulting in a matrix with a size of $\boldsymbol{n_{sim}} \times (\boldsymbol{n} \times \boldsymbol{m})$. In this study, a simulation interval of 48 h was adopted, whereas the results of the last 24 h only were used to eliminate the effect of the initial conditions. The simulation time step was set to 1 min (i.e., $n = 24 \times 60 = 1{,}440$).

The input matrix representing the injection vectors and the output matrix representing the simulation results were merged and saved in an external Pickle file (Van Rossum 2020), in which the first $3n_J$ columns indicate the input of the MLP-NN, whereas the remaining $n \times m$ columns indicate the output. The multiprocessing library in Python (Van Rossum 2020) was used to simultaneously simulate multiple injection events in SWMM, aiming to increase the efficiency of the data set generation process.

Data Set Preprocessing

The input matrix generated in the previous step was first preprocessed before being fed as an input to the MLP-NN. To that end, two transformations were implemented to convert the input matrix into a more representative form that would allow the MLP-NN to learn the true relationship between the input and the output matrices. The first transformation was done by reshaping the injection vector to a concentration time series with the size of $n_{sim} \times 24n_J$. For instance, a subset (512,6,5) of the injection vector was reshaped to a concentration time series as follows:

$$[0,0,0,0,0,512,512,512,512,512,0,0,0,0,0,0,0,0,0,0,0,0,0,0] \tag{4}$$

The second transformation was done by standardizing the input and output matrices by applying the Standard Scaler library (Pedregosa et al. 2011).

MLP-NN Training and Testing

To build the MLP-NN surrogate model, 6,000 random injection events were generated using the previously mentioned approach and simulated using SWMM. Then the input and output matrices were divided into training and testing data sets with a 4:1 ratio using the traintestsplit tool. By utilizing the neural network module in the scikit-learn package in Python (Pedregosa et al. 2011), the training data set was fed to train the surrogate model, while the testing data set was used to verify the surrogate model's accuracy. The structure of the used MLP-NN model was as follows: one hidden layer, two neurons, identity activation function, Adam solver, and 50,000 maximum iterations.

In this study, the surrogate model was meant to produce multiple outputs, by which the simulation time series at one or more observation junctions are represented. Hence, the MLPRegressor library (Pedregosa et al. 2011) was utilized to train an MLP-NN model for each time step at each observation junction, whereas the MultiOutputRegressor library (Pedregosa et al. 2011) was used to join these models to form a complete time series for each observation junction. Accordingly, the number of MLP-NN models composing the surrogate model equals the number of time steps multiplied by the number of observation junctions ($n \times m$). Similar to data set generation, the multiprocessing library was utilized to train several MLP-NN concurrently.

MLP-NN Prediction

New predictions (i.e., simulation results) are typically retrieved from a trained MLP-NN by applying the predict method on a new input data point (i.e., injection vector). However, to improve the prediction speed, weights and biases of the trained MLP-NN models were extracted, and the calculations were conducted explicitly. For the case of one hidden layer, new predictions were calculated by solving the matrix shown in Eq. (5). For the Nth MLP-NN model ($N \in n \times m$), P_N is the output, w_N and b_N are the weights and biases of the hidden layer, ω_N and β_N are these of the output layer, and x is the injection vector. For α neurons, the sizes of ω_N, w_N, b_N, and β_N are $1 \times \alpha$, $\alpha \times 24n_J$, $\alpha \times 1$, and 1×1 respectively

$$P_N = \omega_N[w_N x + b_N] + \beta_N \tag{5}$$

Case Study and Test Scenarios

The developed SI framework was tested on a medium-sized benchmark combined sewer network. Different testing scenarios with a variety of objectives, parameters, and assumptions were examined to reveal the capabilities and limitations of the developed framework.

Case Study Description

The SWMM Example 8 benchmark network (Banik et al. 2017; Sambito et al. 2020; Sambito and Freni 2021) was used as a case study. Network data were retrieved from the SWMM applications manual (Gironás et al. 2009). The combined sewer network shown in Fig. 4 serves an area of 0.117 km^2 and consists of several flow regulation units (e.g., weirs, orifices, a storage unit, and a pump), six subcatchments, 28 junctions, 29 conduits, and two outfalls. Outfall O1 represents the WWTP, while outfall O2 represents an outlet for combined sewer overflows that may occur during the wet flow condition. The observation data are assumed to be collected by an observation sensor placed at the storage well upstream of the pump.

The subcatchments collect stormwater and wastewater from residential and commercial zones. The average daily inflow assigned at each subcatchment for the dry weather flow (DWF) condition is shown in Table 2 (Gironás et al. 2009). In this study, the DWF pattern assigned to the subcatchments (Fig. 5) was assumed to follow the typical wastewater inflow pattern developed by Butler et al. (2018).

The decay of the species of interest was considered to follow first-order kinetics described by Eq. (6), which has been routinely

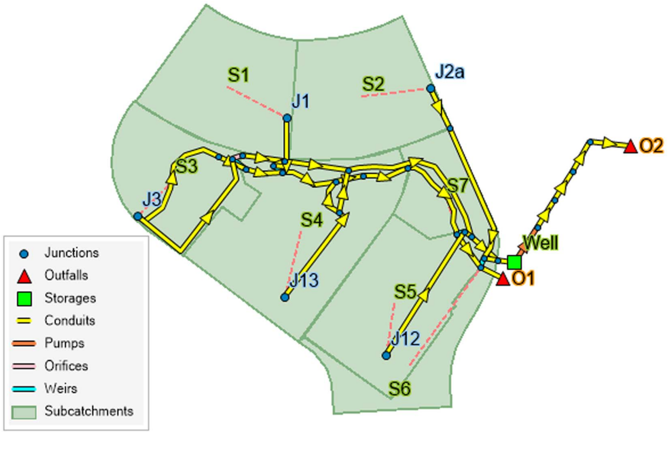

Fig. 4. SWMM Example 8 network layout.

Table 2. Inflow assignments for the DWF condition

Subcatchment	Outlet junction	DWF (L/min)
S1	J1	13.6
S2	J2a	17.0
S3	J3	6.8
S4	J13	20.9
S5	J12	21.2

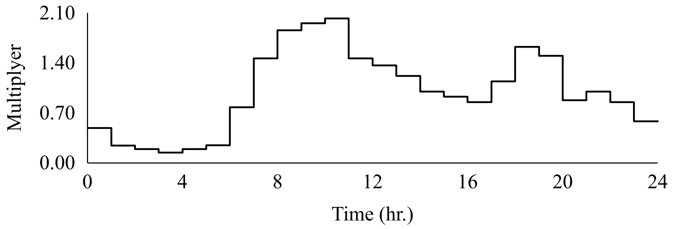

Fig. 5. Diurnal variation of DWF pattern considered in SWMM Example 8.

Table 3. Residence time from each junction to the observation junction

Junction	Residence time (min)
J1	29.94
J2a	15.05
J3	63.47
J13	26.58
J12	11.79

used to describe the decay dynamics of different species in sewer systems (Guo et al. 2022; Lin et al. 2021; Shi et al. 2022)

$$\frac{dC}{dt} = -kC \tag{6}$$

where $(dC)/(dt)$ = rate of change in the injection concentration; k = first-order decay rate coefficient; and C = injection concentration. To use a sensible value for the decay rate coefficient (k), the half-life time of the species of interest ($t_{1/2}$) was normalized by the average residence time (t_R). To calculate t_R, the residence time from each possible source junction to the observation junction was calculated by dividing the pipe lengths by the average flow velocity (Table 3), which produced an average $t_R = 29.36$ min for the case study network.

In this study, two species with different decay rates were tested. The low-decay species has a half-life time ($t_{1/2}$) equal to twice the average residence time (t_R) (i.e., a decay rate of $k = 17$ days^{-1}). In contrast, the high-decay species has a $t_{1/2} = t_R$, resulting in a k of 34 days^{-1}.

Test Scenarios

To investigate the performance and reliability of the proposed SI framework under different conditions, four sets of scenarios were tested. The first set of scenarios (S1) aims to test the model's ability to identify a single injection source within the network. Accordingly, five injection scenarios covering all possible sources (e.g., J1, J2a, J3, J13, and J12) were studied. In S2, the SI framework was used to identify a pair of simultaneous injection sources. Accordingly, 20 injection scenarios were tested, each scenario representing a permutation of two of the five sources. Similarly, the permutation of three injection sources was investigated in sets S3 and S4, resulting in a total of 60 injection scenarios per set. For scenarios with more than one injection source, the injection at each source happens at a different time (Table 4). For example, in S2, the permutation of the two sources [J1,J2] is different from [J2,J1].

As mentioned previously, two different species were used to study the effect of changing the decay rate on the SI framework results. A low-decay species was used in S1-L and S2-L, and a high-decay species was used in S1-H and S2-H. In addition to the decay rate, the SI framework sensitivity to the definition of the actual number of injection sources was also examined. In S1, S2, and S3, the total number of injection junctions (n_i) was defined in the model as one, two, and three sources, respectively. In S4, n_i was not defined in the model. In other words, the SI framework treated all the network junctions as simultaneous injection sources. The goal was to test whether the model can identify nonsource junctions as well as source junctions. One surrogate model was used for each decay rate because the surrogate model was created in a way that allows it to be used for different values of n_i.

Finally, the influence of the injection time during the day was assessed. In Set S1, a species with a concentration of 50 mg/L was injected during the morning low-loading period (low DWF period), which starts at 1:00 a.m. and continues for 5 h (Fig. 5). In S2, an additional injection with the same concentration was assumed to occur during the morning high-loading period (high DWF period), which starts at 7:00 a.m. and lasts for the same period (Fig. 5). In S3 and S4, an additional injection of 50 mg/L occurred in the afternoon medium-loading period (medium DWF period), which starts at 1:00 p.m. and lasts for 5 h (Fig. 5). The summary of all the testing scenarios conducted in this study is listed in Table 4. All scenarios were performed using a workstation computer equipped with an Intel i7-10700 CPU (Intel's HQ, Santa Clara, California) at 4.80 GHz, and 16 GB of RAM.

Results and Discussion

In this section, the results of the proposed SI framework are presented. First, the performance of the surrogate model is validated by comparing its predictions to the SWMM model results. Then the results of the case study testing scenarios are analyzed and discussed.

Surrogate Model

Model Validation

As previously mentioned, a significant number of simulations must be carried out to identify the correct injection event (Table 1).

Table 4. Summary of testing scenarios

Scenario set	n_i	Tested decay rates (days^{-1})	Number of scenarios	Injection characteristics	n_i definition
S1	1	17	5	$C = 50$ mg/L	Defined
		34	5	$H = 1$ a.m.	
				$P = 5$ h	
S2	2	17	20	$C = 50$ mg/L	Defined
		34	20	$H = 1$ and 7 a.m.	
				$P = 5$ h	
S3	3	17	60	$C = 50$ mg/L	Defined
S4			60	$H = 1$ a.m., 7 a.m., and 1 p.m.	Undefined
				$P = 5$ h	

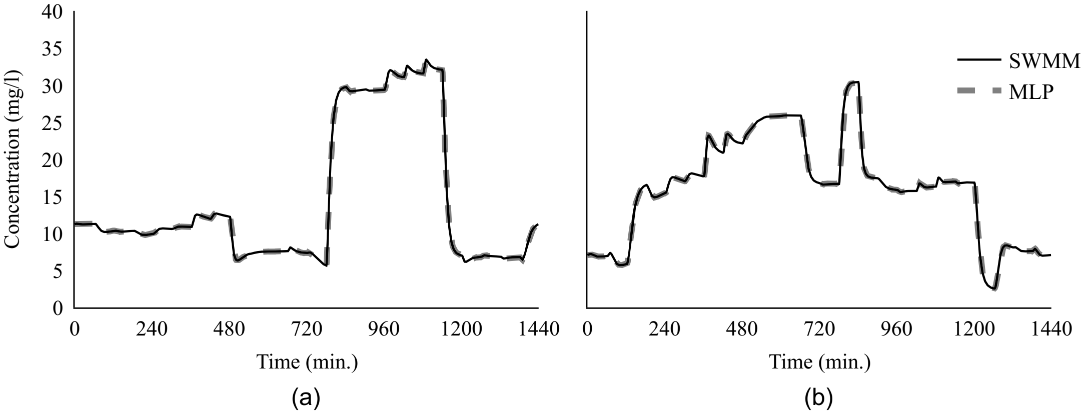

Fig. 6. SWMM model simulation results compared with MLP-NN surrogate model predictions for the surrogate models of (a) low-decay; and (b) high-decay species for a random event.

To enhance the efficiency of the SI framework, an MLP-NN surrogate model was proposed as a simulation tool in place of the SWMM model. Accordingly, comparing the MLP-NN model results with those of the SWMM model is an important step to validate the accuracy of the proposed surrogate model. To that end, 20% of the 6,000 generated injection events were used as a testing data set. The surrogate model displayed remarkable accuracy ($R^2 \cong 1$), which can be clearly seen from Fig. 6. This figure shows the concentration profiles simulated by MLP-NN and SWMM models for the low- and high-decay species. In this figure, the x-axis denotes the time in minutes, the y-axis denotes the observed concentration at the monitoring location, the solid line represents the SWMM model results, and the dotted line represents the MLP-NN model predictions.

Computational Cost Reduction

A single SWMM simulation of the case study network takes 1/15 s on average. Accordingly, the computational cost of generating the 6,000 injection events required for training and testing the surrogate model was ~400 s, while that of training the surrogate model was ~250 s. Thus, the total overhead time needed to generate the data set and train the surrogate model was ~10.8 min. For the GA parameters selected in this study (Table 1) a maximum of 4 million simulations need to be conducted. For the surrogate model, these take approximately 25 min, while for SWMM, the 4 million simulations would take 74 h (neglecting the time taken to perform the GA operators).

Fig. 7 compares the run time performance of the SWMM model-based and surrogate model-based optimization models. The x-axis represents the run time corresponding to the number of simulations on the y-axis. This run time difference highlights the effectiveness of the proposed surrogate-modeling approach. Due to the huge run time reduction and the significant accuracy achieved by the surrogate model, it was used as the simulation tool for the remainder of this study. Similar results were also achieved when both the surrogate model and SWMM were applied to a real-world network three times larger than the case study network (Section S2).

To test whether the computational cost of the SI framework could be further reduced by eliminating the need for the GA optimization module, two backward machine learning–based SI methods were investigated in addition to the forward simulation-optimization approaches presented in this paper. The first inverse approach involved inversely solving the equations of the trained forward MLP-NN to produce the inputs from the outputs, while in the other approach, an MLP-NN model was trained to predict the injection characteristics directly from the observed concentrations. As described in Section S3, these two methods failed to produce satisfactory SI results, and hence only the forward simulation-optimization approaches are featured in this study.

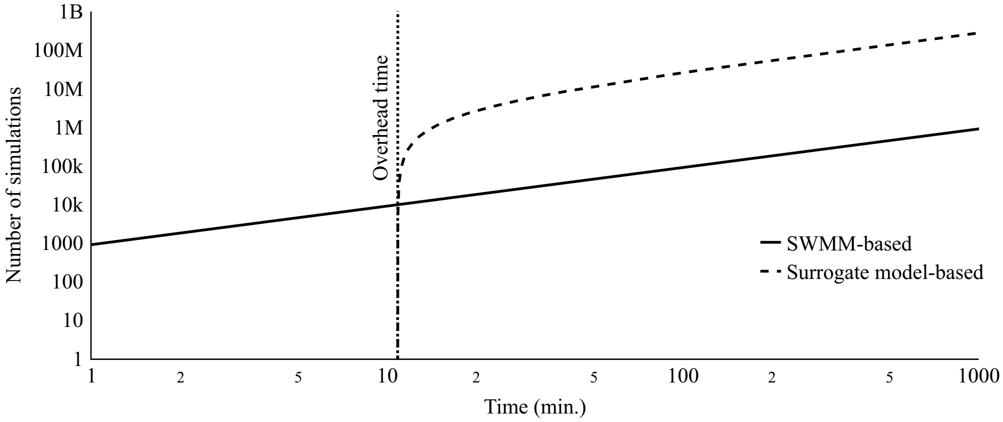

Fig. 7. Run time of the SWMM-based and the surrogate model-based optimization models.

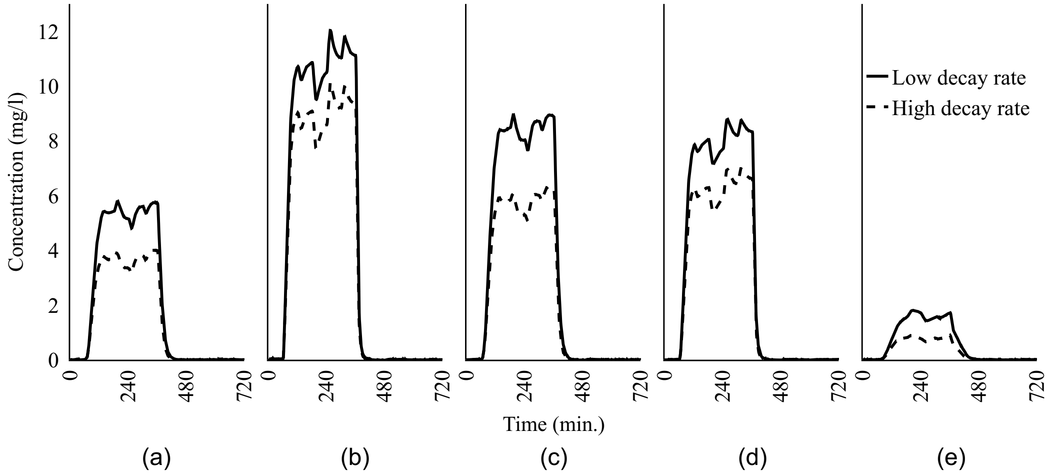

Fig. 8. Response at the monitoring location corresponding to an injection at (a) J1; (b) J12; (c) J13; (d) J2a; and (e) J3 of both low- and high-decay species.

Test Scenarios Results

In this study, four sets of scenarios were conducted to test the SI framework performance (Table 4). In each set, the SI framework was used to identify the location and the characteristics of a different number of simultaneous injection sources under different conditions. In the first set of scenarios (S1), the injection was assumed to take place at one source, whereas Sets S2 and S3 featured two and three simultaneous injection sources, respectively. In S1, S2, and S3, the number of injection sources (n_i) was defined in the model, while for S4, the number of sources was unknown to the model (i.e., the model assumes that all junctions are sources $n_i = n_J$). In Sets S1 and S2, two different cases were examined, namely, low-decay species ($k = 17$ days^{-1}) and high-decay species ($k = 34$ days^{-1}).

Test Scenarios Set 1 (S1)

In S1, the SI framework was able to identify the correct injection source characteristics (i.e., location and pattern) in all 50 optimization runs for all injection scenarios. Thus, the true identification rate (TIR), which represents the percentage of correctly identified sources out of the 50 optimization runs, was equal to 100% for all five injection scenarios. The response produced at the observation location from injecting at each of the possible sources is shown in Fig. 8. In this figure, each panel depicts a different injection location, where the *x*-axis represents the time, and the *y*-axis represents the injection concentration for both low- and high-decay species. For all five injection locations, the optimization process successfully converged to the correct injection characteristics, rendering a response at the observation location that exactly matches the observed pattern.

Test Scenarios Set 2 (S2)

In S2, 20 injection scenarios were tested for both the high- and low-decay species. In each scenario, the injection took place at two locations simultaneously (i.e., a source pair). Herein, the TIR refers to the correct identification percentage of the source pair location (i.e., the percentage of the optimization runs in which the location of both injection sources is correctly identified).

Influence of Decay Rate and Injection Time. Figs. 9(a and b) depict the TIR for each source pair for the low-decay species [Fig. 9(a)] and the high-decay species [Fig. 9(b)]. In these figures, the *x*- and *y*-axes represent the location of the injection source during the high- and low-loading periods, respectively. For the low-decay species [Fig. 9(a)], the SI framework was able to identify the correct source pair with a TIR ≥50% for 17 out of 20 injection scenarios. Similarly, 19 out of 20 injection scenarios were correctly identified with a TIR ≥50% for the high-decay species [Fig. 9(b)]. In addition, the average TIR of the low-decay species was 82.5% compared with 86.5% for the high-decay species. These results indicate that the SI mode performance in identifying the high-decay species is slightly better than the low-decay species. Although this might be counterintuitive, it can be attributed to the fact that the increased decay enhances the uniqueness of the signal produced by the injection sources, and hence improves the detection ability of the SI framework.

Almost all the injection scenarios featuring Junction J3 were easily detected by the SI framework (average TIR ≥98%) for both low- and high-decay species. This can be attributed to the uniqueness of the signal produced by J3 due to its long residence time compared to the other junctions (Table 3). Generally, the residence time appeared to play a significant role in the SI framework's ability to identify the injection location [Figs. 9(c and d)]. Furthermore, the asymmetry of the TIR matrix [Figs. 9(a and b)] indicates that the correct identification of the injection location depends on the injection time within the day (i.e., during the high- or the low-loading period). For example, injection scenarios featuring J1 in the high-loading period generally experienced lower TIRs than those featuring J1 in the low-loading period. Conversely, the TIR for injection scenarios including J12 in the high-loading period were higher than those including J12 in the low-loading period.

Influence of Injection Source Location. To further investigate the performance of the SI framework, we assessed its ability to identify the location of an injection source within a source pair separately (i.e., regardless of the other source). To that end, two identification percentages were calculated: (1) the junction true identification rate (JTIR), which indicates the rate a certain junction is correctly identified regardless of whether its pair was correctly identified or not; and (2) the junction wrong identification rate (JWIR), which represents the rate a certain junction is misidentified with another junction. The summation of the JWIR does not necessarily complement the JTIR to reach 100% because for some injection scenarios neither of the two source junctions is correctly identified.

Figs. 9(c and d) illustrate both the JTIR and JWIR for the low- and high-decay species, where the *x*- and *y*-axes denote the actual and the identified source junction as detected by the SI framework, respectively. The diagonal of the identification matrix represents

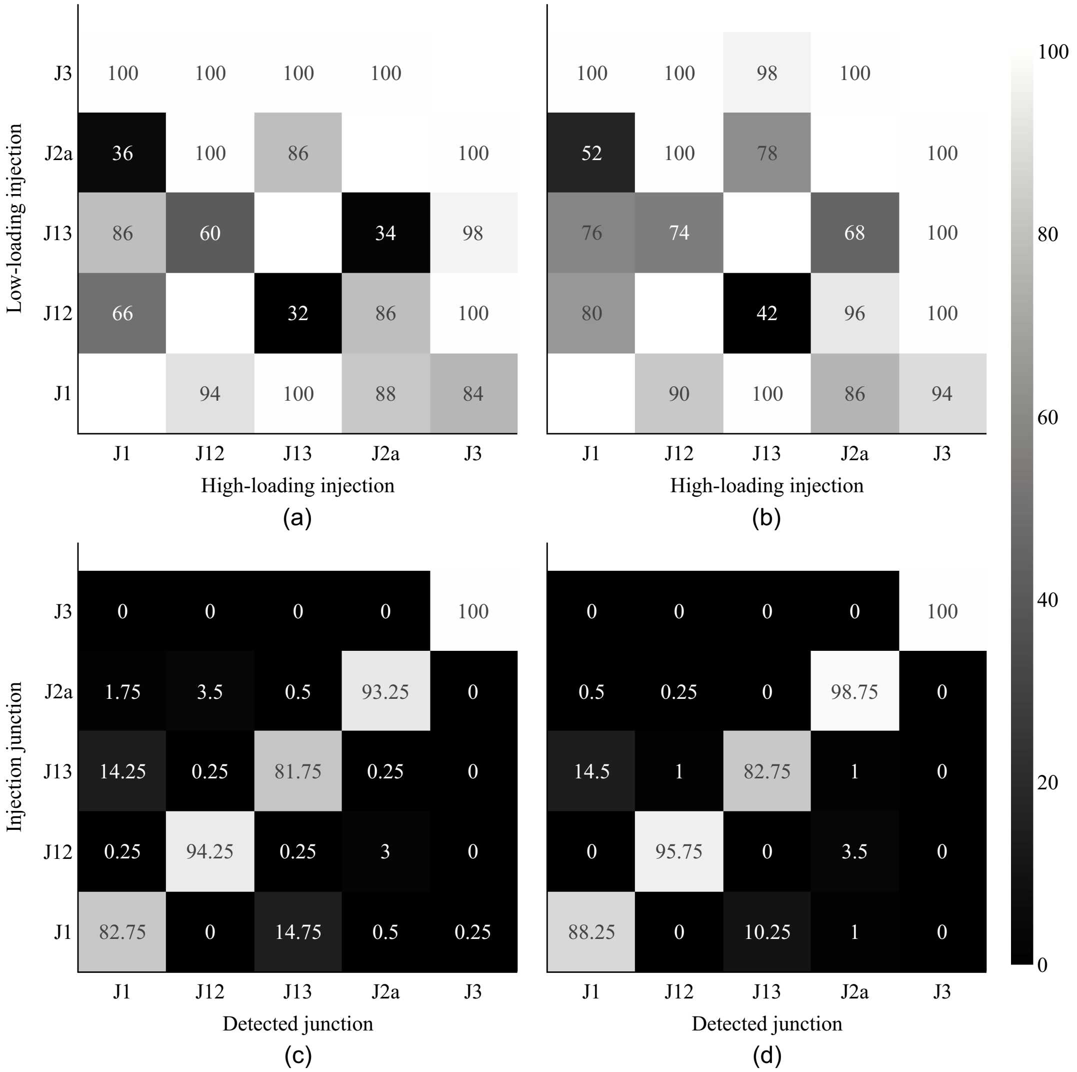

Fig. 9. TIR matrix for the injection scenarios tested in (a) S2-L; and (b) S2-H, and the identification matrix of (c) S2-L; and (d) S2-H, where the diagonals represent the junction true identification rate, and the off-diagonals represent the junction wrong identification rate.

the JTIR, while the off-diagonals represent the JWIR. The analysis showed that J3 was the most correctly identified junction (highest JTIR) for both high- and low-decay species. As explained previously, this can be attributed to the distinct residence time of J3 compared to the other junctions.

On the other hand, J13 was found to be the most misidentified junction (highest JWIR). This is mainly attributed to its consistent confusion with J1, whose residence time is very close to J13 (3.36-min difference). These results further highlight the significance of the residence time as the main driver of the signature induced at the observation junction. In addition, this supports the conclusion that the more unique the residence time from the injection location to the observation location, the more likely the injection junction to be correctly identified.

Objective Function as an Indicator. Typically, the value of the objective function reflects the goodness of the solution. For the presented SI problem, this means that the smaller the value of the objective function, the closer the identified injection characteristics are to the true ones. Herein, we test this hypothesis by examining the relationship between the value of the objective function and the error in the identified injection characteristics (i.e., the SI error). As mentioned previously, the objective function represents the nRMSE between the observed and simulated concentrations at the observation junction. Similarly, the SI error is calculated as the nRMSE between the actual and simulated concentration patterns at the true injection sources.

In Fig. 10, the *x*-axis denotes the value of the objective function, while the *y*-axis denotes the value of the SI error (log scale).

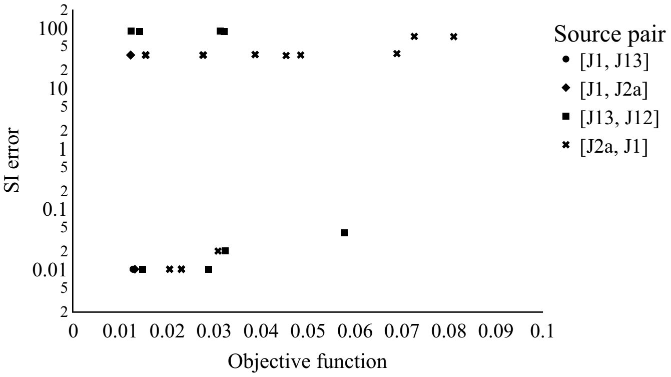

Fig. 10. Relationship between the objective function and the SI error for selected optimization instances in S2-L. Optimization runs with an objective function equal to zero are omitted.

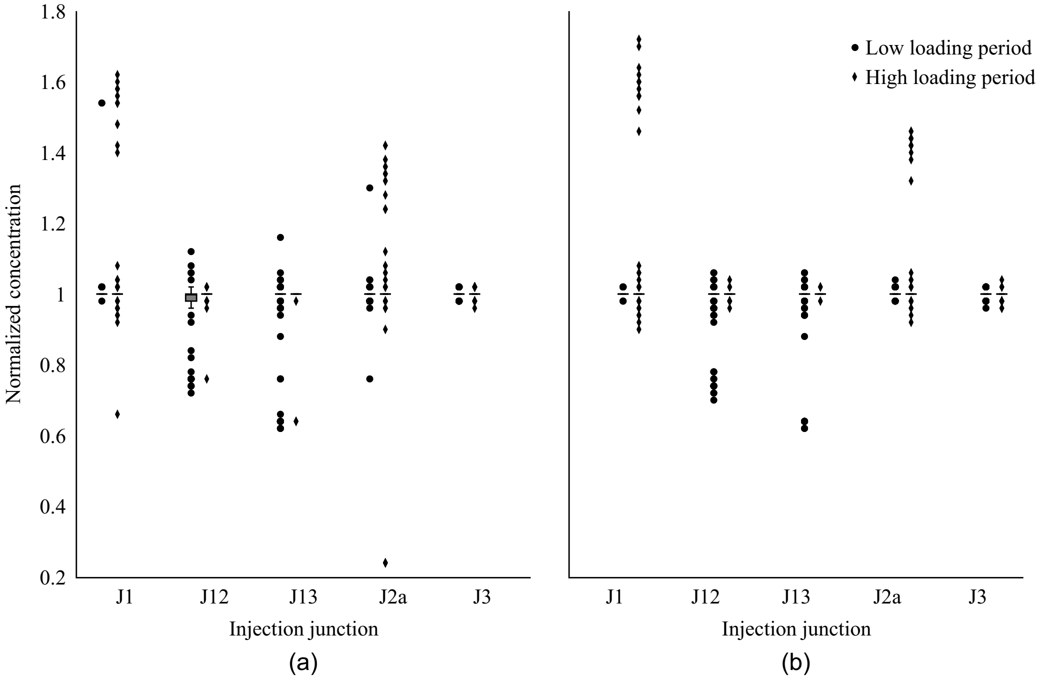

Fig. 11. Normalized concentration results for low- and high-loading periods in (a) S2-L; and (b) S2-H.

The comparison is limited to the nonconverged optimization runs, because the converged runs would yield zero values for the objective function and the SI error. Fig. 10 shows no clear relationship between the value of the objective function and the SI error. For instance, the optimization runs in the top left corner feature small objective function values (0.01–0.03) and large SI errors (30–110). More importantly, the runs featuring the lowest objective function values did not necessarily result in the lowest SI error. For instance, for the pair [J13, J12] represented by squares in Fig. 10, the two runs with the least objective function values had some of the highest SI errors among all runs. A deeper look at these instances revealed that they represent the cases where the injection patterns were correctly identified but for misidentified injection locations.

Taken together, these results highlight the key challenge of the SI problem, which is that different source locations can produce very similar signals at the monitoring location. This means that, for the same observed pattern, there is no one unique solution to the SI problem. Additionally, the signal generated at the observation junction appears to be more sensitive to the injection pattern than the injection location. These results have significant implications for the design of sensor networks. It is crucial to include source identifiability in the criteria used for optimizing sensor placement or the selection of sampling locations.

Injection Pattern Identification. Focusing on the injection concentration of the correctly identified source pairs, the range of the detected concentration by the SI framework was examined for injections during the low- and high-loading periods. The detected concentration was normalized by the actual injection concentration (50 mg/L) and then used to draw the box plot with outliers for S2-L and S2-H (Fig. 11). In this figure, the *x*-axis represents the injection junction, while the *y*-axis shows the normalized concentration detected by the SI framework for this junction for all correctly identified source pairs where this junction was featured. Fig. 11 shows that for both S2-L and S2-H, the median of the normalized concentration of both loading periods is equal to 1 for all junctions. This highlights the high accuracy of the SI framework in correctly identifying the injection concentration. Similar results with comparable accuracy and even higher precision (a smaller number of outliers) were retrieved for the injection start time and duration (results not shown).

Test Scenarios Set 3 (S3)

In S3, the SI framework was able to identify the location of all three injection sources at least once in the 50 optimization runs for all 60 injection scenarios. The average TIR over all 60 scenarios was 47%, compared with 82.5% in S2-L, and the TIR exceeded 50% in 26 of the 60 injection scenarios (i.e., 43% of the scenarios). These results highlight the robustness of the SI framework in identifying multiple injection sources but also demonstrate the difficulty introduced by increasing the number of injection sources. Due to its unique residence time, J3 was featured in 9 out of 10 injection scenarios with the best TIR, generally scoring a TIR between 84% and 100%. This again can be attributed to the unique residence time of J3, which makes it easier to identify compared with other candidate junctions.

Detection Frequency as an Indicator. For 26 out of the 60 injection scenarios in S3, at least one of the 50 optimization runs converged completely to a zero value for the objective function. For 21 of the remaining 34 scenarios, the optimization run with the least objective function value represented the solution with correctly identified source locations.

In addition to the objective function value, we examined whether the frequency of the detected injection sources through all 50 optimization runs can serve as a good indicator for selecting the best optimization run. In Fig. 12, the *x*-axis denotes the 34 nonconverged injection scenarios, and the *y*-axis denotes the detected injection junctions. By analyzing the three most frequently detected junctions, we found that they indeed match the actual injection locations in 15 of 34 nonconverged injection scenarios (marked arrows in Fig. 12).

These results show that both the frequency analysis and the objective function value can be used to judge the quality of the solution obtained by the SI framework, with the objective function

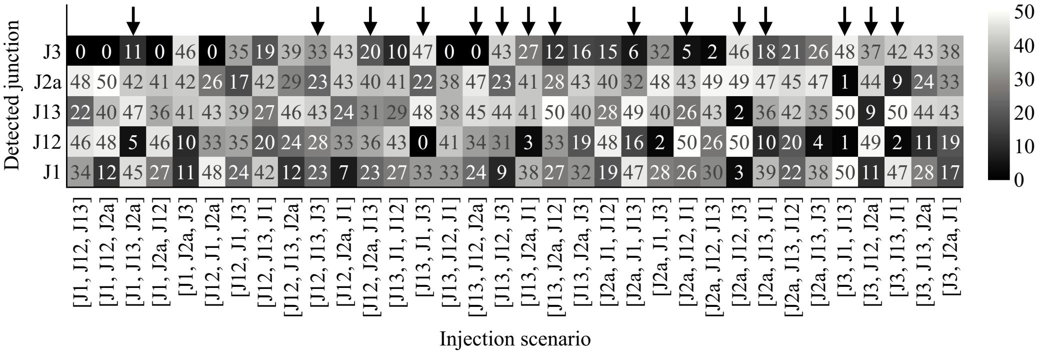

Fig. 12. Junctions' detection frequency for the injection scenarios in S3.

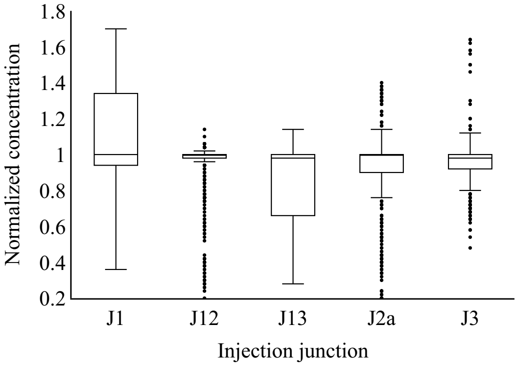

Fig. 13. Normalized concentration results in S3, after dropping the loading period differences.

being a slightly better metric even though it may mislead in some cases (Fig. 10).

Injection Pattern Identification. Similar to S2, the range of the normalized concentrations detected by the SI framework was examined for the correctly identified injection scenarios. Fig. 13 shows a box plot of the normalized detected concentration, where the *x*-axis represents the injection junction and the *y*-axis shows the corresponding normalized concentration, and unlike S2, the loading period was dropped in this figure. By comparing Figs. 11–13, it can be seen that the detection range is much larger in S3. Also, the outliers are more spread out in S3. These results highlight the difficulty faced by the SI framework that resulted from increasing the injection sources to three instead of two. Similar results were retrieved for the injection start time and duration.

Test Scenarios Set 4 (S4)

Surprisingly, in S4, none of the 50 optimization runs completely converged to a zero objective function in all the injection scenarios. In other words, hiding the actual number of injection sources from the model prevented it from converging to the correct solution to the SI problem in all the injection scenarios. Although the actual number of injection sources is the same as S3 (i.e., three sources), the SI framework failed to identify the location and characteristics of the injection sources because it was forced to consider all five junctions as simultaneous injection sources. This can be clearly

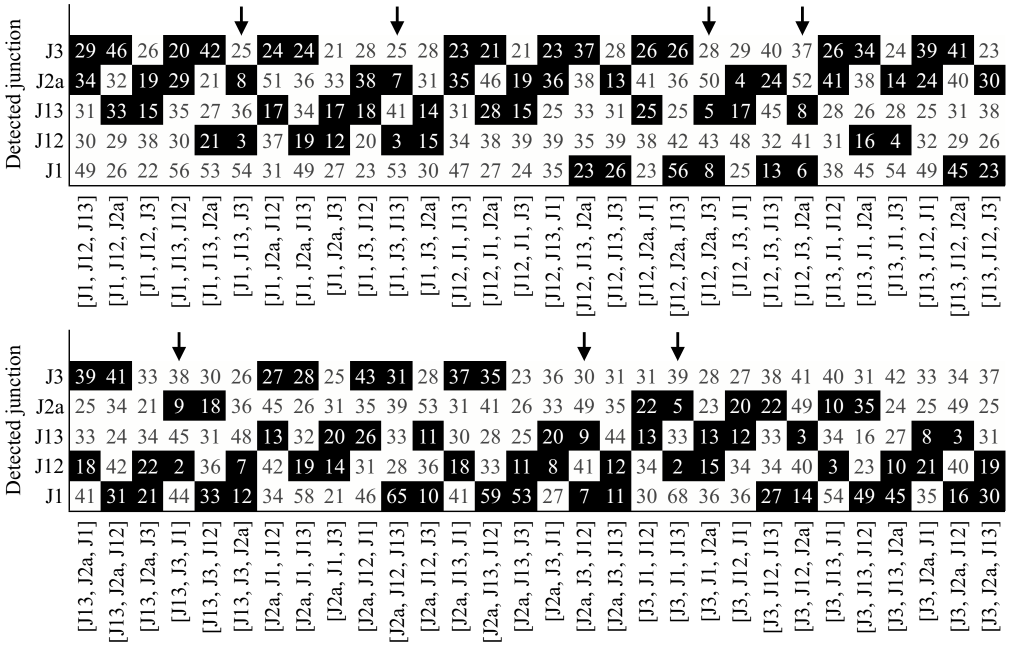

Fig. 14. Average injection concentration at each junction in the network for each injection scenario in S4, where the white cells represent the injection junctions.

seen in Fig. 14, which visualizes the results of S4 in terms of the location of the injection sources. In this figure, the x-axis represents the injection scenarios, and the y-axis shows the detected injection junctions. In Fig. 14, the white cells denote the actual injection junctions, and the number inside each cell represents the average concentration detected by the SI framework. Fig. 14 shows that only seven injection scenarios were identified with low injection concentration error (<10 mg/L) at noninjection junctions (scenarios marked with arrows in Fig. 14).

Taken together, these results highlight the significant challenge of identifying an unknown number of sources. Such scenarios are particularly relevant to WBE applications, where all of the sewer shed junctions must be considered as potential simultaneous sources. In such cases, it may not be enough to collect samples from a single observation location (e.g., WWTP), and additional samples should also be collected from various locations throughout the sewer shed to enhance source identifiability. Furthermore, the proposed source identification algorithm can aid in the selection of the locations for additional sample collection. The latter can be done by designing a sensor placement algorithm that uses source identifiability (i.e., minimizing source identification errors) as an optimization objective.

Conclusions

In this study, a machine learning framework was proposed to solve the source identification problem in sewer systems. Unlike previous studies, the developed framework is capable of identifying multiple simultaneous sources in the sewer shed, as well as accounting for the decay of the species of interest during its transport through the sewer network. The SI problem was formulated as a simulation-optimization problem. To conduct water quality simulations, two different approaches were investigated and compared: SWMM and an MLP-NN surrogate model. The accuracy and computational efficiency of the surrogate MPL-NN model were demonstrated through several comparisons with the SWMM model.

The SI framework was applied to a range of injection scenarios within a case study featuring a midsize combined sewer network. We examined the performance of the proposed framework in identifying the characteristics of different numbers of simultaneous injection sources. In addition, the effect of the species decay rate and the injection period was assessed, along with the impact of identifying the number of injection sources in the framework.

The results of the testing scenarios showed that the proposed framework was able to identify the characteristics of multiple injection sources with remarkable accuracy when the number of sources is defined in the model. In contrast, the proposed framework struggled in detecting the correct injection characteristics when the number of injection sources was unknown, effectively forcing the model to consider all the network junctions as simultaneous sources. Additionally, the results showed that the decay rate and the injection period have a moderate impact on the detectability of the injection sources. On the other hand, the residence time appeared to play a significant role in identifying the correct injection sources, where the source locations possessing the most unique residence time were the most accurately identified by the framework.

Data Availability Statement

All data, models, and codes that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

This work was supported by funding from the University of Illinois Chicago, and the National Science Foundation under Grant No. 2015603.

Notation

The following symbols are used in this paper:

b_N = biases of the hidden layer;
C = injection concentration;
C^{obs} = species concentration observed at the observation junction;
C_{range} = range of injection concentrations;
C^{sim} = species concentration simulated by the simulation model (either SWMM or MLP-NN);
$(dC)/(dt)$ = rate of change in the injection concentration;
i = time step index;
j = index of the observation junction;
k = first-order decay rate coefficient;
m = total number of observation junctions;
n = total number of time steps;
n_i = number of injection sources;
n_J = number of junctions in the network;
n_{sim} = total number of simulation events;
$O_j(S_i)_e$ = simulation result at time i at the observation junction j for the eth simulation event;
P_N = output of the Nth MLP-NN model ($N \in n \times m$);
w_N = weights of the hidden layer;
x = injection vector;
β_N = biases of the output layer; and
ω_N = weights of the output layer.

Supplemental Materials

Sections S1–S3 are available online in the ASCE Library (www.ascelibrary.org).

References

Adedoja, O. S., Y. Hamam, B. Khalaf, and R. Sadiku. 2018. “Towards development of an optimization model to identify contamination source in a water distribution network.” *Water* 10 (5): 1–27. https://doi.org/10.3390/w10050579.

Ahmad, J., M. Ahmad, A. R. A. Usman, and M. I. Al-Wabel. 2021. “Prevalence of human pathogenic viruses in wastewater: A potential transmission risk as well as an effective tool for early outbreak detection for COVID-19.” *J. Environ. Manage.* 298 (Nov): 113486. https://doi.org/10.1016/j.jenvman.2021.113486.

Banik, B. K., L. Alfonso, A. S. Torres, A. Mynett, C. Di Cristo, and A. Leopardi. 2015. “Optimal placement of water quality monitoring stations in sewer systems: An information theory approach.” *Procedia Eng.* 119 (Jan): 1308–1317. https://doi.org/10.1016/j.proeng.2015.08.956.

Banik, B. K., C. Di Cristo, and A. Leopardi. 2014. “SWMM5 toolkit development for pollution source identification in sewer systems.” *Procedia Eng.* 89 (Jan): 750–757. https://doi.org/10.1016/j.proeng.2014.11.503.

Banik, B. K., C. Di Cristo, A. Leopardi, and G. de Marinis. 2017. “Illicit intrusion characterization in sewer systems.” *Urban Water J.* 14 (4): 416–426. https://doi.org/10.1080/1573062X.2016.1176220.

Broad, D. R., G. C. Dandy, and H. R. Maier. 2005. “Water distribution system optimization using metamodels.” *J. Water Resour. Plann.*

Manage. 131 (3): 172–180. https://doi.org/10.1061/(ASCE)0733-9496(2005)131:3(172).

Butler, D., C. J. Digman, C. Makropoulos, and J. W. Davies. 2018. *Urban drainage*. 4th ed. Boca Raton, FL: CRC Press.

Dhama, K., S. K. Patel, M. I. Yatoo, R. Tiwari, K. Sharun, J. Dhama, S. Natesan, Y. S. Malik, K. P. Singh, and H. Harapan. 2021. "SARS-CoV-2 existence in sewage and wastewater: A global public health concern?" *J. Environ. Manage.* 280 (Feb): 111825. https://doi.org/10.1016/j.jenvman.2020.111825.

Diaz-Fierros, T. F., J. Puerta, J. Suarez, and V. F. Diaz-Fierros. 2002. "Contaminant loads of CSOs at the wastewater treatment plant of a city in NW Spain." *Urban Water* 4 (3): 291–299. https://doi.org/10.1016/S1462-0758(02)00020-1.

Di Cristo, C., and A. Leopardi. 2008. "Pollution source identification of accidental contamination in water distribution networks." *J. Water Resour. Plann. Manage.* 134 (2): 197–202. https://doi.org/10.1061/(ASCE)0733-9496(2008)134:2(197).

Elbeltagi, E., T. Hegazy, and D. Grierson. 2005. "Comparison among five evolutionary-based optimization algorithms." *Adv. Eng. Inf.* 19 (1): 43–53. https://doi.org/10.1016/j.aei.2005.01.004.

Gad, A. F. 2021. "PyGAD: An intuitive genetic algorithm Python library." Preprint, submitted June 11, 2021. https://arxiv.org/abs/2106.06158.

Gironás, J., L. A. Roesner, J. Davis, and L. A. Rossman. 2009. *Storm water management model applications manual*. Cincinnati: National Risk Management Research Laboratory, Office of Research and Development, USEPA.

Gromaire, M. C., S. Garnaud, M. Saad, and G. Chebbo. 2001. "Contribution of different sources to the pollution of wet weather flows in combined sewers." *Water Res.* 35 (2): 521–533. https://doi.org/10.1016/S0043-1354(00)00261-X.

Guo, Y., M. Sivakumar, and G. Jiang. 2022. "Decay of four enteric pathogens and implications to wastewater-based epidemiology: Effects of temperature and wastewater dilutions." *Sci. Total Environ.* 819 (May): 152000. https://doi.org/10.1016/j.scitotenv.2021.152000.

Haupt, R. L., and S. E. Haupt. 2003. *Practical genetic algorithms*. Hoboken, NJ: Wiley.

Hou, Z., W. Lao, Y. Wang, and W. Lu. 2021. "Hybrid homotopy-PSO global searching approach with multi-kernel extreme learning machine for efficient source identification of DNAPL-polluted aquifer." *Comput. Geosci.* 155 (Oct): 104837. https://doi.org/10.1016/j.cageo.2021.104837.

Hu, C., J. Zhao, X. Yan, D. Zeng, and S. Guo. 2015. "A MapReduce based parallel niche genetic algorithm for contaminant source identification in water distribution network." *Ad Hoc Networks* 35 (Dec): 116–126. https://doi.org/10.1016/j.adhoc.2015.07.011.

Kessler, A., A. Ostfeld, and G. Sinai. 1998. "Detecting accidental contaminations in municipal water networks." *J. Water Resour. Plann. Manage.* 124 (4): 192–198. https://doi.org/10.1061/(ASCE)0733-9496(1998)124:4(192).

Kim, M., C. Y. Choi, and C. P. Gerba. 2013. "Development and evaluation of a decision-supporting model for identifying the source location of microbial intrusions in real gravity sewer systems." *Water Res.* 47 (13): 4630–4638. https://doi.org/10.1016/j.watres.2013.04.018.

Lee, Y. J., C. Park, and M. L. Lee. 2018. "Identification of a contaminant source location in a river system using random forest models." *Water* 10 (4): 1–17. https://doi.org/10.3390/w10040391.

Li, J., W. Lu, and J. Luo. 2021. "Groundwater contamination sources identification based on the long-short term memory network." *J. Hydrol.* 601 (Oct): 126670. https://doi.org/10.1016/j.jhydrol.2021.126670.

Li, Z., and X.-Z. Mao. 2011. "Global multiquadric collocation method for groundwater contaminant source identification." *Environ. Modell. Software* 26 (12): 1611–1621. https://doi.org/10.1016/j.envsoft.2011.07.010.

Lin, W., Z. Huang, S. Gao, Z. Luo, W. An, P. Li, S. Ping, and Y. Ren. 2021. "Evaluating the stability of prescription drugs in municipal wastewater and sewers based on wastewater-based epidemiology." *Sci. Total Environ.* 754 (Feb): 142414. https://doi.org/10.1016/j.scitotenv.2020.142414.

Liu, L., S. R. Ranjithan, and G. Mahinthakumar. 2011. "Contamination source identification in water distribution systems using an adaptive dynamic optimization procedure." *J. Water Resour. Plann. Manage.* 137 (2): 183–192. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000104.

Lugão, B. C., D. C. Knupp, and P. P. G. W. Rodriges. 2022. "Direct and inverse simulation applied to the identification and quantification of point pollution sources in rivers." *Environ. Modell. Software* 156 (Aug): 105488. https://doi.org/10.1016/j.envsoft.2022.105488.

Majumder, P., and C. Lu. 2021. "A novel two-step approach for optimal groundwater remediation by coupling extreme learning machine with evolutionary hunting strategy based metaheuristics." *J. Contam. Hydrol.* 243 (Dec): 103864. https://doi.org/10.1016/j.jconhyd.2021.103864.

McDonnell, B., K. Ratliff, M. Tryby, J. Wu, and A. Mullapudi. 2020. "PySWMM: The Python interface to Stormwater Management Model (SWMM)." *J. Open Source Software* 5 (52): 2292. https://doi.org/10.21105/joss.02292.

Pan, T.-C., and J.-J. Kao. 2009. "GA-QP model to optimize sewer system design." *J. Environ. Eng.* 135 (1): 17–24. https://doi.org/10.1061/(ASCE)0733-9372(2009)135:1(17).

Panasiuk, O., A. Hedström, J. Marsalek, R. M. Ashley, and M. Viklander. 2015. "Contamination of stormwater by wastewater: A review of detection methods." *J. Environ. Manage.* 152 (Apr): 241–250. https://doi.org/10.1016/j.jenvman.2015.01.050.

Pedregosa, F., et al. 2011. "Scikit-learn: Machine learning in Python." *J. Mach. Learn. Res.* 12 (Oct): 2825–2830.

Perelman, L., and A. Ostfeld. 2013. "Bayesian networks for source intrusion detection." *J. Water Resour. Plann. Manage.* 139 (4): 426–432. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000288.

Preis, A., and A. Ostfeld. 2006. "Contamination source identification in water systems: A hybrid model trees–linear programming scheme." *J. Water Resour. Plann. Manage.* 132 (4): 263–273. https://doi.org/10.1061/(ASCE)0733-9496(2006)132:4(263).

Preis, A., and A. Ostfeld. 2007. "A contamination source identification model for water distribution system security." *Eng. Optim.* 39 (8): 941–947. https://doi.org/10.1080/03052150701540670.

Preis, A., and A. Ostfeld. 2008. "Genetic algorithm for contaminant source characterization using imperfect sensors." *Civ. Eng. Environ. Syst.* 25 (1): 29–39. https://doi.org/10.1080/10286600701695471.

Preis, A., and A. Ostfeld. 2011. "Hydraulic uncertainty inclusion in water distribution systems contamination source identification." *Urban Water J.* 8 (5): 267–277. https://doi.org/10.1080/1573062X.2011.596549.

Rossman, L. A. 2015. *Storm water management model user's manual*. Cincinnati: National Risk Management Research Laboratory, Office of Research and Development, USEPA.

Rossman, L. A., H. Woo, M. Tryby, F. Shang, R. Janke, and T. Haxton. 2000. *EPANET user manual*. Cincinnati: Water Infrastructure Division, Center for Environmental Solutions and Emergency Response, USEPA.

Sambito, M., C. Di Cristo, G. Freni, and A. Leopardi. 2020. "Optimal water quality sensor positioning in urban drainage systems for illicit intrusion identification." *J. Hydroinf.* 22 (1): 46–60. https://doi.org/10.2166/hydro.2019.036.

Sambito, M., and G. Freni. 2021. "Strategies for improving optimal positioning of quality sensors in urban drainage systems for non-conservative contaminants." *Water* 13 (7): 934. https://doi.org/10.3390/w13070934.

Sangkham, S. 2021. "A review on detection of SARS-CoV-2 RNA in wastewater in light of the current knowledge of treatment process for removal of viral fragments." *J. Environ. Manage.* 299 (Dec): 113563. https://doi.org/10.1016/j.jenvman.2021.113563.

Shi, J., X. Li, S. Zhang, E. Sharma, M. Sivakumar, S. P. Sherchan, and G. Jiang. 2022. "Enhanced decay of coronaviruses in sewers with domestic wastewater." *Sci. Total Environ.* 813 (Mar): 151919. https://doi.org/10.1016/j.scitotenv.2021.151919.

Sun, A. Y., S. L. Painter, and G. W. Wittmeyer. 2006. "A robust approach for iterative contaminant source location and release history recovery." *J. Contam. Hydrol.* 88 (3–4): 181–196. https://doi.org/10.1016/j.jconhyd.2006.06.006.

Suthar, S., S. Das, A. Nagpure, C. Madhurantakam, S. B. Tiwari, P. Gahlot, and V. K. Tyagi. 2021. "Epidemiology and diagnosis, environmental resources quality and socio-economic perspectives for COVID-19 pandemic." *J. Environ. Manage.* 280 (Feb): 111700. https://doi.org/10.1016/j.jenvman.2020.111700.

Vankayala, P., A. Sankarasubramanian, S. R. Ranjithan, and G. Mahinthakumar. 2009. "Contaminant source identification in water distribution networks under conditions of demand uncertainty." *Environ. Forensics* 10 (3): 253–263. https://doi.org/10.1080/15275920903140486.

Van Rossum, G. 2020. *The Python library reference, release 3.8.2*. Wilmington, DE: Python Software Foundation.

Wang, C., and S. Zhou. 2017. "Contamination source identification based on sequential Bayesian approach for water distribution network with stochastic demands." *IISE Trans.* 49 (9): 899–910. https://doi.org/10.1080/24725854.2017.1315782.

Wang, H., and K. W. Harrison. 2014. "Improving efficiency of the Bayesian approach to water distribution contaminant source characterization with support vector regression." *J. Water Resour. Plann. Manage.* 140 (1): 3–11. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000323.

Wu, W., J. Ren, X. Zhou, J. Wang, and M. Guo. 2020. "Identification of source information for sudden water pollution incidents in rivers and lakes based on variable-fidelity surrogate-DREAM optimization." *Environ. Modell. Software* 133 (Jul): 104811. https://doi.org/10.1016/j.envsoft.2020.104811.

Xuesong, Y., S. Jie, and H. Chengyu. 2017. "Research on contaminant sources identification of uncertainty water demand using genetic algorithm." *Cluster Comput.* 20 (Jun): 1007–1016. https://doi.org/10.1007/s10586-017-0787-6.

Yang, Y. J., R. C. Haught, and J. A. Goodrich. 2009. "Real-time contaminant detection and classification in a drinking water pipe using conventional water quality sensors: Techniques and experimental results." *J. Environ. Manage.* 90 (8): 2494–2506. https://doi.org/10.1016/j.jenvman.2009.01.021.

Yuan, S., X. Wang, R. Wang, R. Luo, Y. Shi, B. Shen, W. Liu, Z. Yu, and P. Xiang. 2020. "Simultaneous determination of 11 illicit drugs and metabolites in wastewater by UPLC-MS/MS." *Water Sci. Technol.* 82 (9): 1771–1780. https://doi.org/10.2166/wst.2020.445.

Zhu, Y., W. Oishi, C. Maruo, M. Saito, R. Chen, M. Kitajima, and D. Sano. 2021. "Early warning of COVID-19 via wastewater-based epidemiology: Potential and bottlenecks." *Sci. Total Environ.* 767 (May): 145124. https://doi.org/10.1016/j.scitotenv.2021.145124.