sensors

MDPI

Article

Pollution Source Localization in Wastewater Networks

Krystian Chachuła [1], Robert Nowak [1,*] and Fernando Solano [1,2]

[1] Faculty of Electronics and Information Technology, Warsaw University of Technology, 00-665 Warsaw, Poland; k.chachula@tele.pw.edu.pl (K.C.); fs@tele.pw.edu.pl (F.S.)
[2] Blue Technologies, 02-684 Warsaw, Poland
* Correspondence: robert.nowak@pw.edu.pl; Tel.: +48-22-234-7718

Abstract: In December 2016, the wastewater treatment plant of Baarle-Nassau, Netherlands, failed. The failure was caused by the illegal disposal of high volumes of acidic waste into the sewer network. Repairs cost between 80,000 and 100,000 EUR. A continuous monitoring system of a utility network such as this one would help to determine the causes of such pollution and could mitigate or reduce the impact of these kinds of events in the future. We have designed and tested a data fusion system that transforms the time-series of sensor measurements into an array of source-localized discharge events. The data fusion system performs this transformation as follows. First, the time-series of sensor measurements are resampled and converted to sensor observations in a unified discrete time domain. Second, sensor observations are mapped to pollutant detections that indicate the amount of specific pollutants according to a priori knowledge. Third, pollutant detections are used for inferring the propagation of the discharged pollutant downstream of the sewage network to account for missing sensor observations. Fourth, pollutant detections and inferred sensor observations are clustered to form tracks. Finally, tracks are processed and propagated upstream to form the final list of probable events. A set of experiments was performed using a modified variant of the EPANET Example Network 2. Results of our experiments show that the proposed system can narrow down the source of pollution to seven or fewer nodes, depending on the number of sensors, while processing approximately 100 sensor observations per second. Having considered the results, such a system could provide meaningful information about pollution events in utility networks.

Keywords: continuous monitoring; information fusion and sensors; internet of things; multisensor fusion

check for updates

Citation: Chachuła, K.; Nowak, R.; Solano, F. Pollution Source Localization in Wastewater Networks. *Sensors* **2021**, *21*, 826. https://doi.org/10.3390/s21030826

Academic Editor: Vassilis S. Kodogiannis
Received: 15 December 2020
Accepted: 21 January 2021
Published: 26 January 2021

1. Introduction

In recent years, there has been a growing global concern regarding the security of water distribution systems (WDSs) and wastewater networks (WWNs). WDSs and WWNs are spatially diversified, pervasive, and linked to the basic needs of human society. They are therefore considered as critical infrastructures by all national security agencies.

Events occurring in these systems that can have an impact on civilians include the following:

- Accidental contamination of a WDS leading to contamination as a result of non-potable water surrounding pipe breaks and leaks, or from the back-flow of polluted water from customer facilities.
- Intentional contamination of WDSs by terrorists, i.e., the deliberate poisoning of a given population downstream.
- Prohibited connections to storm water networks that could potentially cause pollution of natural water bodies.
- Careless dumping of waste over WWNs, which could lead to explosions and cause major catastrophes due to the constant presence of flammable gasses produced by existing bacteria.

- Discharge of toxic substances into a WWN, which may result in the release of illegal and harmful concentrations of pollution into the environment.

1.1. Case Studies

There have already been cases of intentional contamination of WDSs by terrorist. Water-related terrorist activities have been reported in ancient Rome, in the United States during the Civil War, in Europe and Asia during World War II, and during the Kosovo conflict of 1999 [1].

With regard to WWNs, the discharge of sulfuric acid (H_2SO_4) to sewers could originate from applications, such as etching of semiconductors, accumulator acid or the production of organic chemical substances [2]. Sodium hydroxide (NaOH) is widely used for cleaning surfaces in metal processing in industrial applications [3], whereas discharge of sodium sulfate (Na_2SO_4) can be caused by the regeneration of cation exchange resins, which are used for softening water in industrial water treatment [4]. Illegal discharge of such dangerous harsh industrial waste into sewage networks could be harmful for the biological stage of waste water treatment plants (WWTPs), its personnel, sewer pipes, and the general public. Once of the most recent cases of this occurred in December 2016, when the WWTP of Baarle-Nassau, Netherlands, failed [5]. The operator noticed that the biological treatment stage failed completely as the pH level in the aeration tank was extremely acidic, with a pH level of nearly 1. This damage was caused by the improper disposal of large volumes of wastewater containing high concentrations of sulfuric acid into the sewage system.

1.2. Past Works

During the past decade, to mitigate the effects of potential polluting events in water systems, the research and industrial communities have focused primarily on three lines of research and development: (1) innovative sensor technologies for monitoring pollutant levels; (2) network planning solutions aimed at providing optimal network coverage constrained to a given capital, expenditure or event likelihood; and (3) source localization methods for detecting the most likely injection point of a pollutant if such an event occurs.

In recent years, several project initiatives [6,7] and prototypes of sensor systems [8–22] for wastewater monitoring have been proposed and studied. These include the design of sensors (electrochemical sensors, optical sensors, mass spectrometry, ion spectrometry, etc.) for manholes, main sewer lines, water bodies, and basins at the WWTP for estimating the presence or concentration of specific pollutants at the point and time of measurement. These systems are not capable of inferring the localization in the network where the pollutant was introduced.

A second line of research involves the design of planning methods for the deployment of a set of sensors in a given network [1,23–27] so that the arrangement of the deployed sensors maximizes the likelihood of detecting any anomaly. Furthermore, some research was done on the topic of portioning WDSs. Di Nardo et al. in [28] proposed a methodology that combines an algorithm for the automated creation of district metered area (DMA) boundaries with practical criteria for DMA design. Ciaponi et al. in [29] focused on proving the benefits of partitioning by simulating a discharge of cyanide and investigating the influence of district isolation on the security of a water supply system.

In this article, we focus on the third line of research: localization methods for detecting the most likely injection point of a pollutant. We present and evaluate a data fusion framework that aids the localization of the most likely source of pollution for sewer networks. The data fusion framework processes measurements collected by point-detection sensors in the sewage network (as input) and it estimates (as output) the likelihood that a sewage network inlet was the source of the pollutant.

In 2008, Di Cristo and Leopardi proposed an iterative procedure for identifying the source of pollution among a set of nodes that are monitored by sensors in a WDS [30]. Di Cristo and Leopardi identified the most likely source of the pollution by solving an optimization problem. The problem formulation minimizes the squared difference between the values measured by a sensor and the hydraulic model values for each node, where the hydraulic conditions of the network allowed for pollution. However, Di Cristo and Leop-ardi did not consider the localization of the pollution source outside the set of monitored nodes.

In the same year, Preis and Ostfeld proposed a genetic algorithm (GA) for solving a similar optimization problem [31]. However, the objective function was formulated as the least-squares difference between the detected (at the monitoring stations) and simulated contaminated values. The GA evaluated different permutations of four problem variables: (1) the contaminant injection node (integer), (2) the injection start time (real), (3) the injection duration (real), and (4) the injection mass rate (real). Two additional studies considered the usage of a GA to solve similar objective functions with the same four problem variables. In [32], the minimization of the absolute value of the difference between the values measured by a sensor and the hydraulic model values was proposed. A year later the minimization of the normalized square difference between simulated and measured contaminant concentration values was depicted [8].

In 2009, Huang and McBean provided provided a heuristic solution to the problem using a different approach [33]. Huang and McBean assumed that the insertion time of a pollutant into the network was known then, the heuristic determined whether a measurement corresponded with the insertion of the pollutant at the source by comparing the arrival times of the measurement to a monitored node with the expected arrival time window estimated by a hydraulic model. By considering a sequence of measurements over all nodes in the network, Huang and McBean estimated the probability that such an injection event was caused at a given node.

In 2010, Sanctis et al. presented the contamination status algorithm (CSA), which is based on the particle backtracking algorithm (PBA) [34]. The PBA infers the mass concentration ratio that every output node in a network shall receive over time from any of its upstream (input) nodes as a linear function. The CSA categorizes the state of the input nodes as safe, unsafe, or unknown based on the concentration ratios over all feasible input–output pairs of nodes in a network derived by the PBA.

All previously mentioned studies provide a methodology for localizing the source of a pollutant injection in a WDS. To the best of our knowledge, the work presented by the authors in this article is the first one inferring on the localization of a pollutant injection in sewage WWN.

In addition to the anomaly localization problem in WDSs and WWNs, there is the anomaly detection problem—the source of an anomaly cannot be found even if an abnormal time series of measurements occurs. Support vector machine (SVM) approaches for anomaly detection are widely used [35–39], but these are not effective at detecting a gradual anomalous change of sensor values in a time-series [35]. Numerous studies have used artificial neural networks (ANNs) for anomaly detection [35,38,40–43].

In the present study, we propose an algorithmic solution that assumes the following:

- the network topology is known and static,
- the localization of the sensor devices is known and static,
- the number of sensor devices is limited and not all points of the sewage network are monitored,
- the sensor devices have heterogeneous but complementary sensing capabilities, and
- the sensor devices sample water quality at a subset of network junctions at arbitrary sampling times.

This article is organized as follows. Section 2 describes the data fusion strategy, Section 3 presents the process and the outcomes of its evaluation, and Section 4 contains the conclusion of our findings.

2. Methods

The sewage network is represented by a directed acyclic graph $G(V, E)$. A node $v \in V$ represents a sewage network junction or spot, such as a building or sewage well. One or more sensors could be deployed in a node. Each edge $e \in E$ represents a pipe between two nodes. The attribute o_e of every edge e provides the current flow propagation time offset (lag) it introduced between its two connecting nodes. The direction of an edge corresponds to the direction of the wastewater flow. Additionally, it is assumed that (1) the graph G is consistent, (2) each node is connected to the root by exactly one path, and (3) the graph G contains a node representing the sink (drain). The sewage network's sink is the location where all the wastewater exits the network. Hereinafter, we use the terms "sink" and "root" interchangeably. The graph G is a directed tree created under the assumptions mentioned above. Section 3 shows two examples of such networks, where root nodes are marked as "1".

The sensors provide measurements of the wastewater properties in the form of observations O [44]:

$$O = \langle Q, v, t, y, \Delta y \rangle, \text{ where} \tag{1}$$

Q is the entity, v is the spatial location of the measurement, t is the time-stamp of the measurement, y is a digital representation of the measured value, and Δy is the uncertainty. Possible entity values for Q include electrical conductance, and pH and concentration of a specific compound. The spatial location of an observation O corresponds to the node v in G where the measurement was taken.

We define the vector of all observed entities as $\boldsymbol{Q} = \langle Q_1, Q_2, \cdots, Q_N \rangle$.

In the presented system a finite list of substances (compounds) to be tracked is represented by set $\boldsymbol{C} = \{C_1, C_2, C_3, \cdots, C_M\}$, where $C_i \in \boldsymbol{C}$ is a compound. Predefined functions are used to convert measured values to amounts for each compound C_i. It should be noted that $\boldsymbol{C}$ included not only pollutants, but also other compounds, primarily those that are generally present in wastewater.

The data fusion algorithm consists of five steps: resampling, pollution quantification, downstream propagation, tracking, and event generation (Figure 1). These steps are repeated. The input data for resampling (first step of the algorithm) consists of sensor measurements, the output data of the resampling is the input for the pollution quantification, etc.

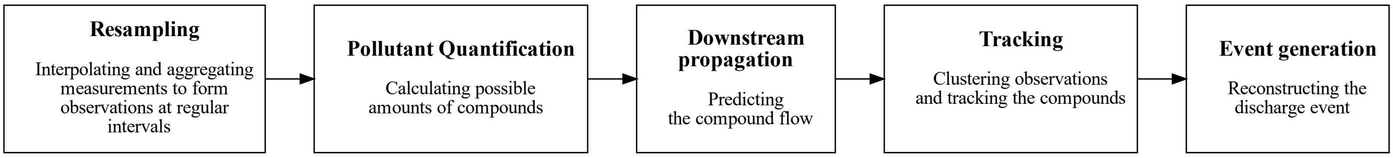

Resampling
Interpolating and aggregating measurements to form observations at regular intervals
Pollutant Quantification
Calculating possible amounts of compounds
Downstream propagation
Predicting the compound flow
Tracking
Clustering observations and tracking the compounds
Event generation
Reconstructing the discharge event

Figure 1. The data fusion algorithm.

2.1. Resampling

Each sensor in the system is capable of sampling at a different time period. In the resampling step, we convert sensor measurements into sensor observations in a unified discrete-time domain by setting a common sampling time period for all sensors and estimating the value of sensor measurements that were not initially collected. Therefore, for each iteration of the data fusion algorithm, the values of y and Δy are calculated. This process utilizes linear interpolation when the sampling period T is greater than the sensor measurement period or if there are missing measurements, and mean aggregation, when the sampling period is less than the sensor measurement period.

Sensor observations resulting from this step are represented as indicated below.

$$O' = \langle Q, v, k, y, \Delta y \rangle, \text{ where} \tag{2}$$

k is the discrete-time step $k = 0, 1, 2, \cdots$.

Time steps are referenced to a fixed point in time t_0 so that measurements taken at t_0 have $k = 0$. In the present study, we assume uniform time sampling. Therefore, discrete-time step k represents $t_k = t_0 + k \cdot T$.

2.2. Pollutant Quantification

The pollution quantification step converts sensor observations $O' = \langle Q, v, k, y, \Delta y \rangle$ into the identification and quantification of sought compounds.

The pollution quantification step yields a set of *pollution detections* $D = \langle D^1, D^2, \cdots, D^M \rangle$. Each pollution detection D^i takes the form $D^i = \left\langle C_i, v, k, a^i, \Delta a^i \right\rangle$, where a^i is the amount in liters of a substance C_i that is detected with uncertainty Δa^i by node v.

Pollution detections are created using the following method. For each sensor observation O', every compound $C \in \boldsymbol{C}$ is considered independently. A potential discharge amount is calculated using the mapping function $f(C, y, \Delta y) \to \langle C, a, \Delta a \rangle$, where y is the measured value of entity Q, C is the compound, and a is the amount.

A threshold value σ is considered for filtering out sensor observations that are below the noise level. In other words, only if the inferred pollution detection amount a^i is greater than the threshold σ, pollution detection is created and added to the detection set. The algorithm used to calculate pollution detections is depicted in Algorithm 1.

```
Algorithm 1: Pollution quantification algorithm
   Input: observations $\boldsymbol{O}'$ in discrete time, $O' = \langle Q, v, k, y, \Delta y \rangle$
   Output: detections $\boldsymbol{D}$, $D = \left\langle c_i, v, k, a^i, \delta a^i \right\rangle$
 1 $\boldsymbol{D} = []$;
 2 for $O'$ in $\boldsymbol{O}'$ do
 3     for $C$ in $\boldsymbol{C}$ do
 4         $D := \mathrm{map}(O', C)$;
 5         if $a^i \geq \sigma$ then
 6             $\boldsymbol{D} := \boldsymbol{D} + D$;
 7         end
 8     end
 9 end
```

The $map(O', O)$ function in line 4 of Algorithm 1 compound amount a from an input sensor observation y in the following way. Let $z = |y - b|$, where y is a sensor observation value and b is the baseline, which is defined as the sensor observation value when no compound is present in the proximity of the sensor. In the presented study, we assume linear mapping from z to the compound amount, $a = \alpha z$. Parameter α specifies how a unit amount of a compound can be quantified into pollutant volume units.

A new detection object is created only if amount a exceeds the detection threshold. Thresholds are set per compound and are constant in time. These thresholds allow us to filter out insignificant detected amounts caused by small fluctuations of measured values.

2.3. Downstream Propagation

The downstream propagation step infers additional pollution detections in vertices of the graph where no sensors are installed. The majority of vertices are like this as we assume the number of available sensors to be limited, due to either high capital or operational costs.

The inferred time of arrival of a pollutant is generated from pollution detections $\boldsymbol{D}$ using Algorithm 2.

A depth-first search algorithm is used to create new detections downstream from their original nodes with maximum depth d, for each pollution detection $D \in \boldsymbol{D}$.

New pollution detections are inferred by considering the propagation model of compounds in the utility network between neighbor vertices. For this, we consider the following three simplifications.

1. The propagation time of a substance for an edge, o_e, is known, constant in time, and equal for every compound. In practice, this condition is satisfied only when the flow characteristics do not change in time and the flow rate for each compound is the same.
2. The total amount of a discharged compound does not change as the substance flows through the network. In practice, a substance may either react with other domestic waste and change its intrinsic characteristics, or may adhere to the sewage pipe walls.
3. The sensors have infinite resolution and no noise. Therefore, tiny volumes of diluted compounds in the network over time can be measured.

Algorithm 2: Detection propagation algorithm

```
Input: detections $\boldsymbol{D}$, $D = \langle C, v, k, a, \Delta a \rangle$
Output: detections $\boldsymbol{D}'$, $D' = \langle C, v, k, a, \Delta a \rangle$
$\boldsymbol{D}' = []$;
for $D$ in $\boldsymbol{D}$ do
    $N :=$ depth-first-search$(G(V, E), v_D, d)$;
    for $n$ in $N$ do
        $D' =$ calculate$(D, n)$;
        if $a_D \geq threshold$ then
            $\boldsymbol{D}' := \boldsymbol{D}' + D$;
        end
    end
end
```

Inferred detections D' with amounts less than a given threshold (representing the process noise) are not considered. After detection propagation, pollution detections are associated with almost every vertex in the graph.

2.4. Tracking

The tracking algorithm clusters pollution detections by the detected compound. A cluster of pollution detections associated with a detected compound is named a *track*. Therefore, pollution detections can not be associated with a track if there is a difference in compounds between the detection and the track.

In this article, a Kalman filter is used for predicting the most probable location of a detected compound within the network in a previous algorithm iteration (for time t_{i-1}). The tracking algorithm updates the most probable location for time t_i using the pollution detections calculated in the previous two steps.

The filter state (Equation (3)) represents the location in the network of a substance at a given point in time. For each track, the most probable amount a of a compound, as well as the most probable location d (as a function of time), are determined. Location is expressed as a real number equal to the distance from the network sink.

$$\hat{x}_{k|k} = \begin{bmatrix} a & d \end{bmatrix}^{\mathrm{T}} \tag{3}$$

The precise location of a compound within a track can be calculated at any time based on the fact that only one path connects each node to the sink and that the starting node of the track is stored. This localization scheme places a compound on the graph edge located at vector $\langle u, v, \alpha \rangle$, where u, v are the source and the destination of the edge, respectively, and $\alpha \in [0, 1)$ is a number describing the position relative to the edge.

The tracking algorithm (Algorithm 3) assigns pollution detections to tracks, creates new tracks, and removes stale tracks. Once the location of compounds within tracks are predicted by the Kalman filter, an assessment of whether the pollution detection can be supported is performed by comparing the amounts a_d, a_t and graph distance D_g between the detections and the track representatives. If these values are less than their respective thresholds, the detection is counted as supporting the track. If the detection cannot be associated with the existing tracks, a new track is created. Tracks with no new associations over several previous algorithm iterations are labeled as outdated and are removed.

Algorithm 3: Tracking algorithm

```
Input: detections D and D′
Output: tracks T (groups of detections)
T := tracks from previous step;
for t in T do
    update(Kalman);
end
for d in {D, D′} do
    for t in T do
        r = representative(t);
        if $c_r = c_d \wedge |a_r - a_d| < th_a \wedge D_g(r, d) < th_g$ then
            t.support := t.support + d;
        end
    end
    if d does not support any t then
        $T := T + \text{init}(d)$;
    end
end
for t in T do
    if $t.support < th_T$ then
        $T := T - t$;
    end
end
```

2.5. Event Generation

The final step, namely, *event generation*, only considers tracks that have a large number of supporting (associated) detections. Events are created for each of these tracks, where an event represents the discharge of a compound into a node of the graph.

Event generation is depicted in Algorithm 4.

Algorithm 4: Event generation algorithm

```
Input: tracks T (groups of detections)
Output: events E
T′ = filter(T);
for t in T do
    e = traverse( t, G(V,E) );
    E := E + e;
end
cluster(E);
sort(E);
```

Possible events are generated for each important track. Subsequently, equivalent events from different paths are transformed to a single event with the confidence being equal to the sum of the confidences of those events and the compound amount being equal to the maximum of the amounts in a cluster. Finally, the events are sorted in descending order of confidence.

2.6. Implementation

The data fusion algorithm was implemented as a Python package with a modular layout. This enabled the user to replace any of the modules with ones that were more suited to their specific use. This was especially important as the subsequent modules of the system were then simplified compared to the real world. The sampling process relied on linear interpolation and mean aggregation. The amounts of the compounds were assumed to be in a linear relationship with the values measured. The detection and clustering thresholds and Kalman filter parameters were constant.

A client-server application was developed to store measurements and implement data fusion. This application also contains a presentation layer that allows the results to be presented using a web browser. Our application use a PostgreSQL database, Python standard packages, and the Django web framework.

3. Results

The fusion algorithm was tested using simulated data. Several numerical experiments were conducted to evaluate our system across multiple scenarios.

Two network topologies were considered. The first network $G_1(V, E)$ was a path graph. It consisted of linearly connected nodes (Figure 2).

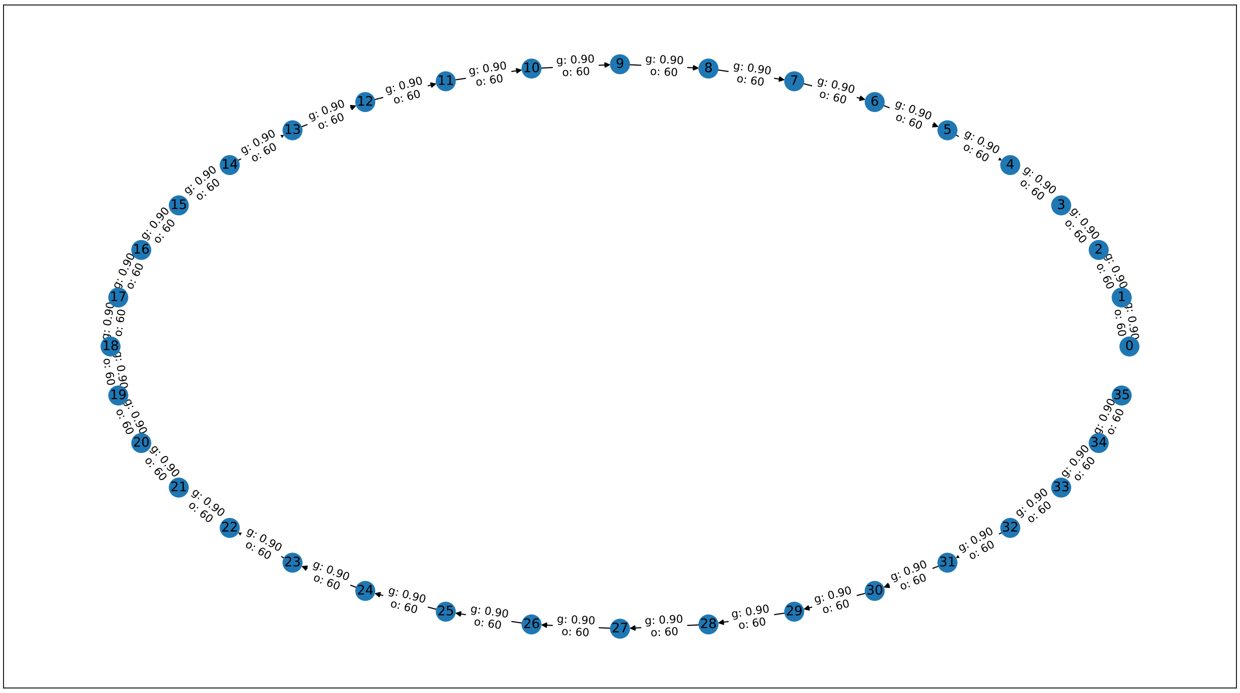


Figure 2. The $G_1(V, E)$ network topology used in simulations.

The second network $G_2(V, E)$ (Figure 3) was a simplified version of the sample network available in the EPANET software. The original network was a water distribution network, so the edges were reversed to resemble a sewage network. The modification included transforming the acyclic graph into a tree via a depth-first search starting at node 1. The edge gains (g_e) and offsets (o_e) were calculated using pipe lengths (l_e) from the original EPANET network description (Equations (4) and (5)). The offsets were computed by dividing the pipe length by a constant velocity $v = 10\frac{km}{h}$.

$$o_e = \frac{l_e}{v} \tag{4}$$

The gains were calculated for each edge using a linear function. The minimal gain $g_{min} = 0.65$ was chosen to make similar travel times for G_1 and G_2.

$$g_e = [l_e - \min(l_e)] \cdot \frac{g_{min} - 1}{\max(l_e) - \min(l_e)} + 1 \tag{5}$$

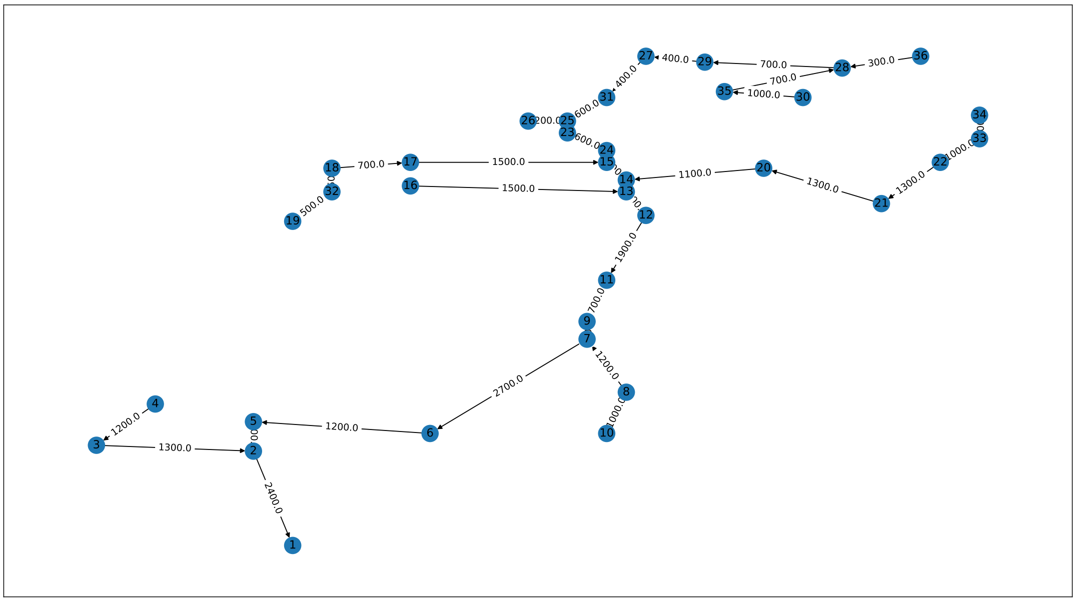


Figure 3. The $G_2(V, E)$ network topology used in simulations (based on the EPANET sample network 2).

We simulated two types of sensors: (1) the microMole sensor system [6] and (2) liquid chromatography with tandem mass-spectrometry (LC-MS-MS). The microMole sensor system measures the pH and electrical conductivity (EC) of wastewater every second. It can be mounted in main sewer pipes of no less than 250 mm in diameter. The microMole system is not capable of identifying chemical compounds. LC-MS-MS is laboratory equipment capable of detecting and quantifying chemical compounds. Within the H2020 SYSTEM project [7], LC-MS-MS is used for analysis of wastewater samples collected at WWTPs. It analyses the composition of wastewater every 10 min. As LC-MS-MS is located at the WWTP, LC-MS-MS data are not sufficient for localizing the source of pollution in a sewage network graph.

Our sensors measured one of three entities of different characteristics:

Q_1 with a range of $[0, +\infty)$ and a neutral value of 1400, which refers to the electrolytic conductivity,

Q_2 with a range of $[0, 14]$ and a neutral value of 7.65, which refers to the pH, and

Q_3 with a range of $[0, 1]$ and a neutral value of 0, which indicates the relative concentration of a pollutant.

The substances (compounds) that were tracked are listed in Table 1. The illegal substance is sodium hydroxide, described in Section 1.1. Pipe cleaner is legal but has a similar pH and electrolytic conductivity. The presence of those compounds in the proximity of the sensors measuring Q_1 and Q_2 affected readings in the same way: a positive peak of Q_1 (Q_1+) and a negative peak of Q_2 (Q_2-). The measured values of Q_3 were influenced only by C_2 in the form of a positive peak (Q_3+).

Table 1. Substances tracked by the data fusion system.

Short	Substance	Legality	pH	EC [mS/cm]
C_1	Pipe cleaner	Legal	12	22–26
C_2	Sodium hydroxide, NaOH	Illegal	12	1

Thresholds for Algorithms 1, 2 and 4, and the Kalman filter parameters were constant for all simulations. The specific values were derived using the expectation-maximization algorithm on a representative sample of the measured values.

The average results of four experiments are presented below. For each network topology, the influence of sensor coverage, substance discharge amount, update period, and downstream propagation depth was calculated. The parameter values are presented in Table 2.

Table 2. Parameters used in numerical experiments.

Parameter	Default Value	Considered Values
Update period [s]	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Sensor coverage	0.5	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1
Discharge amount [l]	25	25, 50, 75, 100, 125, 150, 175, 200
Downstream propagation depth [nodes]	1	1, 2, 3

The updated period T was the constant time period used in resampling that determined how many iterations of the data fusion algorithm were performed.

The sensor coverage was represented by a number in the range $[0, 1]$, which expressed the number of sensors in the network relative to the number of nodes. For a given simulation, Q_1 and Q_2 sensors were placed randomly among all nodes (except the sink) using sampling without replacement. A single sensor measuring Q_3 was always located in the sink of the network.

3.1. Simulations

To create random scenarios for the data fusion module, a measurement generation module was produced. The method for creating simplified sensor observations was conceived based on the results of real-world experiments.

To create a simulation scenario, several parameters of the discharge event were required: the compound, node, amount, noise, and function inverse to the mapping function described in Section 2.2. An additional edge e parameter known as gain g_e was also required. The edge gain g_e was a real number that satisfied $g_e \in [0, 1]$. The amplitude of the signal measured at the edge end divided by the amplitude of the signal measured at the edge start yielded g_e. The gain parameters revealed how the signal was attenuated while the compounds traveled through the edges. Noise was introduced by adding random values from a Gaussian distribution with a mean of 0 and a standard deviation equal to the product of the measurement value and the noise parameter.

Real-world measurements often resemble exponential functions with bases in the range from 0 to 1. In our experiments, a rectangle impulse function was used to simplify reverse mapping of the amount of the compound. Generating a single measurement series consisted of an initial calculation of the target area between the baseline reading and the measured values, and then the generation of a suitable number of measurements. A series corresponding to a single discharge event differed only in signal length (which was calculated by dividing the target area by the product of the gains of all the edges from the discharge node to the current node) and the initial signal amplitude (which was a property of the entity).

During one discharge event, many measurement series were generated that corresponded to each sensor in each node on the path from the discharge node to the sink. Scenarios in which more than one discharge event occurred were not taken into account, as this would have required knowledge of the behavior of compounds when they mix in the sewage network.

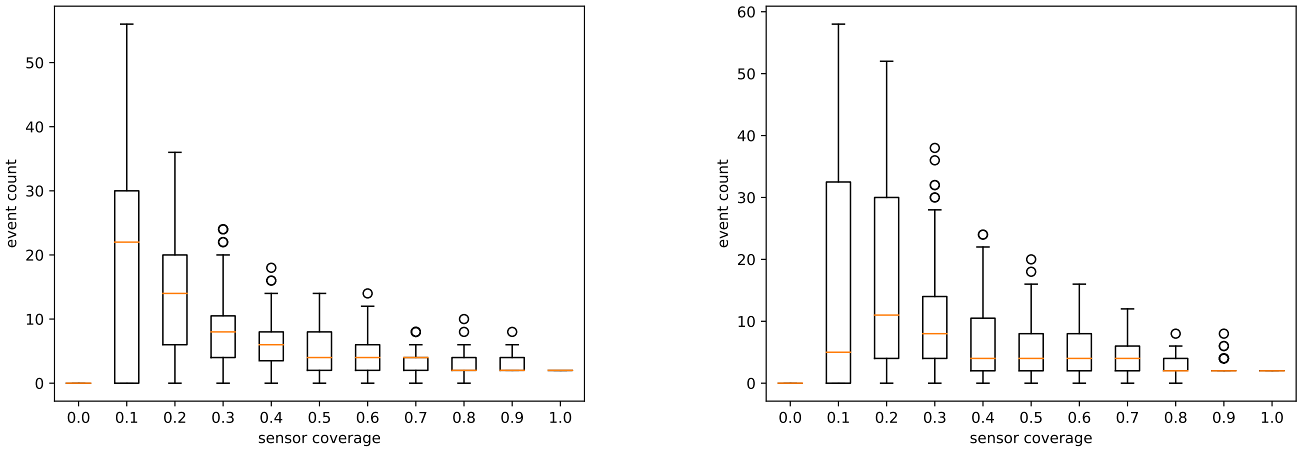
3.2. Quality of Data Fusion

For each scenario, a set of events was generated by the system using many iterations of the fusion algorithm depicted in Figure 1. Scenarios involved the simulated data of a single discharge in a random node, a node where pollution was introduced into the network was selected using a uniform distribution. The detected events were labeled either true positive or false positive. It should be noted that at most, one event was true positive.

Based on these labels, metrics were calculated for each simulation:

- The confidence coefficient, which was computed by dividing the confidence of the true positive event by the average confidence of all events. This metric showed how the confidences of true positive events compared to the confidences of false positive events. For the system to be useful, this metric had to be greater than 1.
- The number of reported events. The ground truth was 1. The smaller this number was, the more precise the localization. In studied scenarios, multiple events signified multiple possible nodes of discharge or multiple compounds; therefore, this was a valuable metric that demonstrated the precision of the system.

The results demonstrated that, as expected, the performance of the system depended on the sensor coverage of the network. According to Figure 4, the number of generated events decreased faster with an increase in the number of sensors in the network. In the case of the simple "path" network (G_1), the event count plot showed a median count of approximately 20 for a coverage of 10%. Taking into account that the a priori knowledge included two similar compounds, the system should reduce the source of the pollution to approximately 10 nodes. A coverage of 20% provides a twofold decrease in the event count.

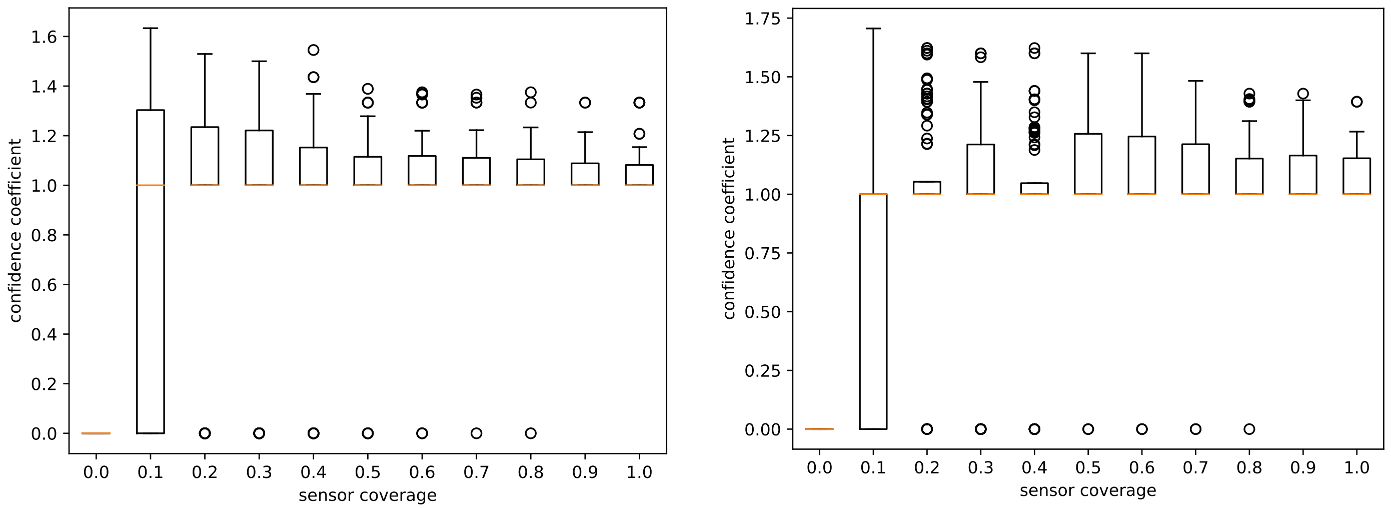


(**a**) Path network (**b**) Net2 network

Figure 4. Event count by sensor coverage. The number of false positives rapidly decreased with an increase in sensor coverage.

According to simulations performed on the more realistic "Net2" network (G_2), the event count should not exceed 15 for coverage of 10% or greater. Moreover, taking into account that two similar compounds were considered, an event should be reduced to approximately seven nodes. Achieving such coverage in real networks may not be possible, but this metric provides a valuable overview of what can be expected concerning the performance of the system. It is important to note that to expect reconstructed events in a single node, sensor coverage would have to reach 100%, which in practice is impossible to achieve. This fact, however, does not mean that one cannot obtain accurate results from the proposed system at low sensor coverage. This means that the lower this value is, the more nodes must be considered as a potential source of pollution.

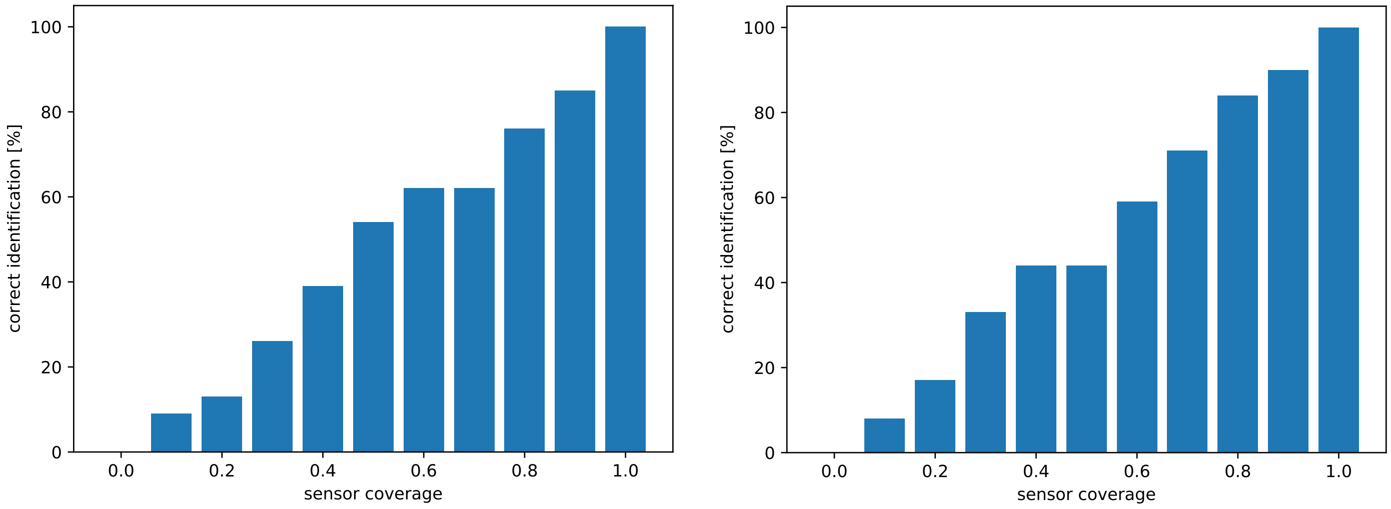
Figure 5 shows that if the position of the sensors and the discharge node are aligned in a way that allows for any detection (confidence coefficient $\neq 0$), assuming that coverage is greater than 10%, it can be expected that true events have greater confidence than false events. Across all experiments with coverage of greater than 10%, true events had $\approx$30% greater confidence than false events in the simple network and $\approx$50% greater confidence in the more complicated network.



(**a**) Path network (**b**) Net2 network

Figure 5. Confidence coefficient by sensor coverage. True positive events had confidence greater than the average confidence across all events.

When it comes to correct identification of the source node, Figure 6 shows that we can get very close to 100% identification chance with network coverage of 80%.



(**a**) Path network (**b**) Net2 network

Figure 6. Percentages of identification of the right source node. Accuracy grows with an increase in sensor coverage.

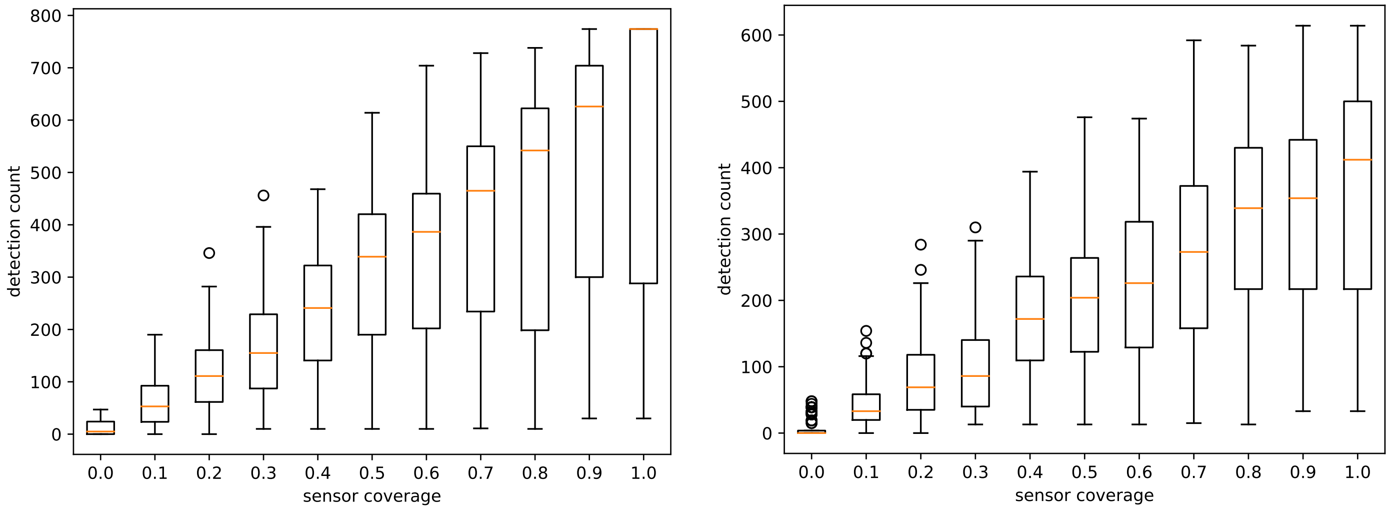
Comparing these results with events counts shown in Figure 4, we can observe that even though a high coverage is needed to achieve excellent accuracy, results achieved at lower coverage would still be useful. When the number of sensors is low and the sensors are located far from the pollution source, we can expect that the system would generate several similar events in multiple nodes in proximity of the source. It is difficult to distinguish the actual source in such a case. However, as already mentioned (Figure 5), on average, our system assigns higher confidence values to actual pollution sources than to neighbor nodes.

Di Cristo and Leopardi in 2008 achieved a location identification rate from 60.5% to 100% with 31% of nodes containing sensors [30], while our system needed 60% coverage to get such high values. However, in the cited article, sensor locations were constant across simulations and only one discharge node was considered. In contrast, we considered random placement of sensors and the discharge node was chosen at random. Additionally, Di Cristo and Leopardi used hydraulic simulation by the EPANET simulator, which causes the performance of the system to be dependent on simulation quality. Our aim was to create a system which could continuously monitor the network and perform calculations as new measurements appear.

3.3. System Performance

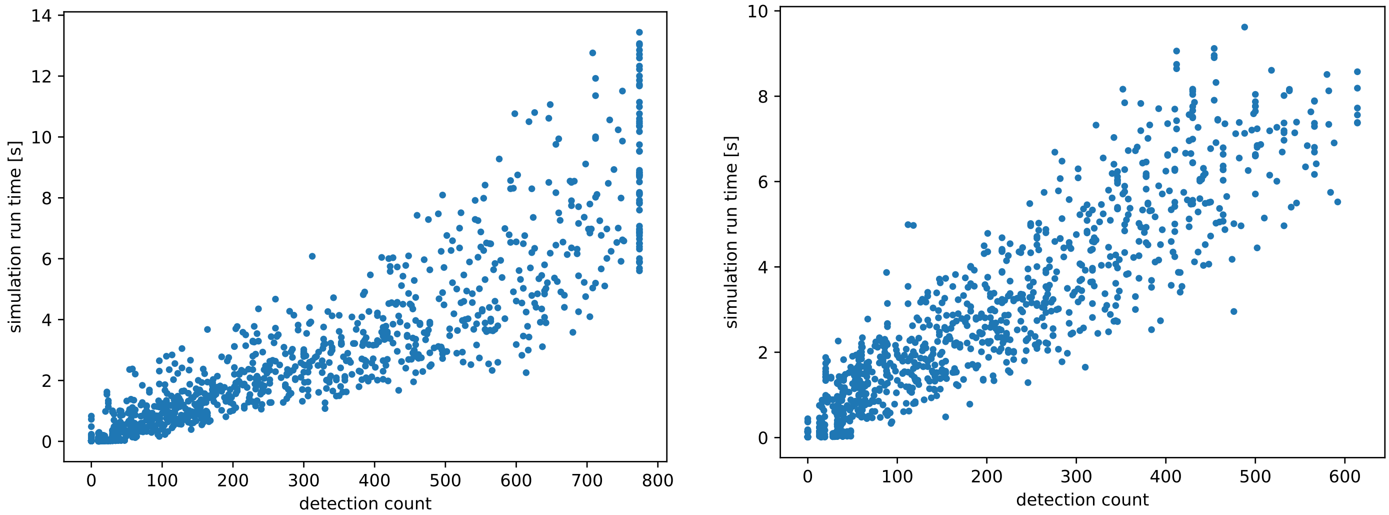
The performance of the system was also evaluated by analyzing the algorithm execution time and the total number of observations that translated into the usage of system memory.

Figure 7 illustrates that memory usage (observation count) was directly proportional to the number of sensors. Analysis of the simulation run time charts (Figure 8) showed that the time complexity of the used algorithms was linear relative to the number of measurements that generated detections. As shown in Figure 8, the system can be expected to process more than 100 sensor observations per second.



(**a**) Path network (**b**) Net2 network

Figure 7. Detection count by sensor coverage. The number of observations increased linearly with coverage.



(**a**) Path network (**b**) Net2 network

Figure 8. Calculation run-time by the number of detections. The time of execution was directly proportional to the total number of observations.

4. Conclusions

The proposed localization strategy allowed the source of the pollution to be pinpointed to a small number of nodes in the networks. Our system correctly detected nearly 100% of events when sensors were present in at least 10% of the network nodes. To achieve such high location identification rates we needed 60% coverage. A large number of sensors were not required to be placed in the network to achieve meaningful results. The algorithms presented in this study can be expected to process at least 100 sensor observations per second.

Further research will focus on model formation for acyclic graphs, not only trees. This will allow us to consider all possible flow paths, therefore improving the quality of the results of data fusion. An additional benefit of this future approach will be the ability to use an unmodified network model in our system, thus removing the need for additional decisions regarding the modification process. Moreover, we will implement of parallel processing to increase the observation processing rate. The research will also include the use of machine learning algorithms in the event generation stage [45] to achieve improved quality compared to the algorithm based on threshold that is currently in use. Last, we plan to consider the uncertainty of measurements by incorporating this into the confidence coefficient of generated events.

Author Contributions: K.C. developed the algorithms, built and tested the software, and performed numerical experiments. R.N. provided the concept, algorithms, and methodology. F.S. formulated the scientific problems and acquired funding. All the authors discussed the results, wrote the original draft, edited and reviewed text, and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the H2020 SYSTEM [7] project, which have received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 787128. This work was supported by Statutory Research of Institute of Computer Science Warsaw University of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Acknowledgments: The manuscript was improved by the anonymous reviewers. We would also like to thank all colleagues from SYSTEM for valuable comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ostfeld, A.; Uber, J.G.; Salomons, E.; Berry, J.W.; Hart, W.E.; Phillips, C.A.; Watson, J.P.; Dorini, G.; Jonkergouw, P.; Kapelan, Z.; et al. The Battle of the Water Sensor Networks (BWSN): A Design Challenge for Engineers and Algorithms. *J. Water Resour. Plan. Manag.* **2008**, *134*, 556–568. [CrossRef]
2. Hauser, F.M.; Hulshof, J.W.; Rößler, T.; Zimmermann, R.; Pütz, M. Characterisation of aqueous waste produced during the clandestine production of amphetamine following the Leuckart route utilising solid-phase extraction gas chromatography-mass spectrometry and capillary electrophoresis with contactless conductivity detection. *Drug Test. Anal.* **2018**, *10*, 1368–1382.
3. Cormier, G.J. Alkaline Cleaning. In *Surface Engineering*; Cotell, C.M., Sprague, J.A., Smidt, F.A., Eds.; ASM International: Materials Park, OH, USA, 1994; pp. 18–20. [CrossRef]
4. Ghasemipanah, K. Treatment of ion-exchange resins regeneration wastewater using reverse osmosis method for reuse. *Desalin. Water Treat.* **2013**, *51*, 5179–5183. [CrossRef]
5. Emke, E.; Vughs, D.; Kolkman, A.; de Voogt, P. Wastewater-based epidemiology generated forensic information: Amphetamine synthesis waste and its impact on a small sewage treatment plant. *Forensic Sci. Int.* **2018**, *286*, e1–e7. [CrossRef]
6. Micromole. Micromole—Sewage Monitoring System for Tracking Synthetic Drug Laboratories. Available online: http://www.micromole.eu (accessed on 17 October 2019).
7. SYSTEM. H2020 SYSTEM—SYnergy of Integrated Sensors and Technologies for Urban sEcured environMent. Available online: https://cordis.europa.eu/project/rcn/220304/factsheet/en (accessed on 25 January 2021).

8. De Vito, S.; Fattoruso, G.; Esposito, E.; Salvato, M.; Agresta, A.; Panico, M.; Leopardi, A.; Formisano, F.; Buonanno, A.; Delli Veneri, P.; et al. A Distributed Sensor Network for Waste Water Management Plant Protection. In *Sensors*; Andò, B., Baldini, F., Di Natale, C., Marrazza, G., Siciliano, P., Eds.; Springer: Cham, Switzerland, 2018; pp. 303–314.
9. Hoes, O.; Schilperoort, R.; Luxemburg, W.; Clemens, F.; van de Giesen, N. Locating Illicit Connections in Storm Water Sewers Using Fiber-Optic Distributed Temperature Sensing. *Water Res.* **2009**, *43*, 5187–5197. [CrossRef] [PubMed]
10. Lim, J.S. Mobile Sensor Network to Monitor Wastewater Collection Pipelines. Ph.D. Thesis, UCLA, Los Angeles, CA, USA, 2012.
11. Lepot, M.; Makris, K.F.; Clemens, F.H. Detection and quantification of lateral, illicit connections and infiltration in sewers with Infra-Red camera: Conclusions after a wide experimental plan. *Water Res.* **2017**, *122*, 678–691. [CrossRef] [PubMed]
12. Tan, F.H.S.; Park, J.R.; Jung, K.; Lee, J.S.; Kang, D.K. Cascade of One Class Classifiers for Water Level Anomaly Detection. *Electronics* **2020**, *9*, 1012. [CrossRef]
13. Tashman, Z.; Gorder, C.; Parthasarathy, S.; Nasr-Azadani, M.M.; Webre, R. Anomaly Detection System for Water Networks in Northern Ethiopia Using Bayesian Inference. *Sustainability* **2020**, *12*, 2897. [CrossRef]
14. Zhang, D.; Heery, B.; O'Neil, M.; Little, S.; O'Connor, N.E.; Regan, F. A Low-Cost Smart Sensor Network for Catchment Monitoring. *Sensors* **2019**, *19*, 2278. [CrossRef]
15. Perfido, D.; Messervey, T.; Zanotti, C.; Raciti, M.; Costa, A. Automated Leak Detection System for the Improvement of Water Network Management. *Proceedings* **2016**, *1*, 28. [CrossRef]
16. Rojek, I.; Studzinski, J. Detection and Localization of Water Leaks in Water Nets Supported by an ICT System with Artificial Intelligence Methods as a Way Forward for Smart Cities. *Sustainability* **2019**, *11*, 518. [CrossRef]
17. Ji, H.; Yoo, S.; Lee, B.J.; Koo, D.; Kang, J.H. Measurement of Wastewater Discharge in Sewer Pipes Using Image Analysis. *Water* **2020**, *12*, 1771. [CrossRef]
18. Kuchmenko, T.A.; Lvova, L.B. A Perspective on Recent Advances in Piezoelectric Chemical Sensors for Environmental Monitoring and Foodstuffs Analysis. *Chemosensors* **2019**, *7*, 39. [CrossRef]
19. Pisa, I.; Santín, I.; Vicario, J.; Morell, A.; Vilanova, R. ANN-Based Soft Sensor to Predict Effluent Violations in Wastewater Treatment Plants. *Sensors* **2019**, *19*, 1280. [CrossRef] [PubMed]
20. Drenoyanis, A.; Raad, R.; Wady, I.; Krogh, C. Implementation of an IoT Based Radar Sensor Network for Wastewater Management. *Sensors* **2019**, *19*, 254. [CrossRef]
21. Ma, J.; Meng, F.; Zhou, Y.; Wang, Y.; Shi, P. Distributed Water Pollution Source Localization with Mobile UV-Visible Spectrometer Probes in Wireless Sensor Networks. *Sensors* **2018**, *18*, 606. [CrossRef]
22. Desmet, C.; Degiuli, A.; Ferrari, C.; Romolo, F.; Blum, L.; Marquette, C. Electrochemical Sensor for Explosives Precursors' Detection in Water. *Challenges* **2017**, *8*, 10. [CrossRef]
23. Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J.; Glance, N. Cost-Effective Outbreak Detection in Networks. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '07, San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: New York, NY, USA, 2007; pp. 420–429. [CrossRef]
24. Banik, B.; Alfonso, L.; Torres, A.; Mynett, A.; Di Cristo, C.; Leopardi, A. Optimal Placement of Water Quality Monitoring Stations in Sewer Systems: An Information Theory Approach. *Procedia Eng.* **2015**, *119*, 1308–1317. [CrossRef]
25. Ostfeld, A.; Salomons, E. Optimal Layout of Early Warning Detection Stations for Water Distribution Systems Security. *J. Water Resour. Plan. Manag.* **2004**, *130*, 377–385. [CrossRef]
26. Propato, M. Contamination Warning in Water Networks: General Mixed-Integer Linear Models for Sensor Location Design. *J. Water Resour. Plan. Manag.* **2006**, *132*, 225–233. [CrossRef]
27. Banik, B.; Alfonso, L.; Di Cristo, C.; Leopardi, A. Greedy Algorithms for Sensor Location in Sewer Systems. *Water* **2017**, *9*, 856. [CrossRef]
28. Di Nardo, A.; Di Natale, M.; Guida, M.; Musmarra, D. Water network protection from intentional contamination by sectorization. *Water Resour. Manag.* **2013**, *27*, 1837–1850. [CrossRef]
29. Ciaponi, C.; Murari, E.; Todeschini, S. Modularity-based procedure for partitioning water distribution systems into independent districts. *Water Resour. Manag.* **2016**, *30*, 2021–2036. [CrossRef]
30. Di Cristo, C.; Leopardi, A. Pollution Source Identification of Accidental Contamination in Water Distribution Networks. *ASCE J. Water Resour. Plan. Manag.* **2008**, *134*, 197–202. [CrossRef]
31. Preis, A.; Ostfeld, A. Genetic algorithm for contaminant source characterization using imperfect sensors. *Civ. Eng. Environ. Syst.* **2008**, *25*, 29–39. [CrossRef]
32. Khan, M.A.I.; Banik, B. Contamination Source Characterization in Water Distribution Network. *Glob. Sci. Technol. J.* **2017**, *5*, 44–55.
33. Huang, J.; McBean, E. Data Mining to Identify Contaminant Event Locations in Water Distribution Systems. *ASCE J. Water Resour. Plan. Manag.* **2009**, *135*, 466–474. [CrossRef]
34. Sanctis, A.E.D.; Shang, F.; Uber, J.G. Real-Time Identification of Possible Contamination Sources Using Network Backtracking Methods. *J. Water Resour. Plan. Manag.* **2010**, *136*, 444–453. [CrossRef]
35. Inoue, J.; Yamagata, Y.; Chen, Y.; Poskitt, C.M.; Sun, J. Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 1058–1065. [CrossRef]

36. Tian, Z.X.; Jiang, J.P.; Guo, L.; Wang, P. Anomaly detection of Municipal Wastewater Treatment Plant operation using Support Vector Machine. In Proceedings of the International Conference on Automatic Control and Artificial Intelligence (ACAI 2012), Xiamen, China, 3–5 March 2012; pp. 518–521. [CrossRef]
37. Jalal, D.; Ezzedine, T. Decision Tree and Support Vector Machine for Anomaly Detection in Water Distribution Networks. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020; pp. 1320–1323. [CrossRef]
38. Garmaroodi, M.S.S.; Farivar, F.; Haghighi, M.S.; Shoorehdeli, M.A.; Jolfaei, A. Detection of Anomalies in Industrial IoT Systems by Data Mining: Study of CHRIST Osmotron Water Purification System. *IEEE Internet Things J.* **2020**, 1. [CrossRef]
39. Ayadi, A.; Ghorbel, O.; Bensaleh, M.S.; Obeid, A.; Abid, M. Outlier detection based on data reduction in WSNs for water pipeline. In Proceedings of the 2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 21–23 September 2017; pp. 1–6. [CrossRef]
40. Fehst, V.; La, H.C.; Nghiem, T.D.; Mayer, B.E.; Englert, P.; Fiebig, K.H. Automatic vs. Manual Feature Engineering for Anomaly Detection of Drinking-Water Quality. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '18, Kyoto, Japan, 15–19 July 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 5–6. [CrossRef]
41. Macas, M.; Wu, C. An Unsupervised Framework for Anomaly Detection in a Water Treatment System. In Proceedings of the 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 1298–1305. [CrossRef]
42. Joslyn, K.; Lipor, J. A Supervised Learning Approach to Water Quality Parameter Prediction and Fault Detection. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 9–12 December 2018; pp. 2511–2514. [CrossRef]
43. Sonrani, M.; Abbatangelo, M.; Carmona, E.; Duina, G.; Malgaretti, M.; Comini, E.; Sberveglieri, V.; Prasad Bhandari, M.; Bolpagni, D.; Sberveglieri, G. Array of Semiconductor Nanowires Gas Sensor for IoT in Wastewater Management. In Proceedings of the 2018 Workshop on Metrology for Industry 4.0 and IoT, Brescia, Italy, 16–18 April 2018; pp. 68–71. [CrossRef]
44. Mitchell, H.B. *Multi-Sensor Data Fusion: An Introduction*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2007.
45. Nowak, R.; Misiurewicz, J.; Biedrzycki, R. Automatic Adaptation in Classification Algorithms Fusing Data From Heterogeneous Sensors. In Proceedings of the 14th IEEE Conference on Information Fusion (FUSION), Chicago, IL, USA, 5–8 July 2011; pp. 1993–1999. Available online: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5977619 (accessed on 25 January 2021).