water

MDPI

Article

A Two-Stage Model for Data-Driven Leakage Detection and Localization in Water Distribution Networks

Vineet Tyagi †, Prerna Pandey *,†, Shashi Jain † and Parthasarathy Ramachandran †

Department of Management Studies, Indian Institute of Science, Bangalore 560012, India; vineet.tyagi@gmail.com (V.T.); shashijain@iisc.ac.in (S.J.); parthar@iisc.ac.in (P.R.)

* Correspondence: pandey.prerna3121@gmail.com

† These authors contributed equally to this work.

Abstract: Water utilities face the challenge of reducing water losses by promptly detecting, localizing, and repairing leaks during their operational stage. To address this challenge, utilities are exploring alternative approaches to detect leaks with high accuracy in a timely manner, while minimizing environmental and economic consequences. This research proposes a two-stage model that relies on data analysis to predict leak incidents and their specific locations in water distribution networks (WDNs). By leveraging pressure and flow rate data collected from multiple points in the network, the model first calculates prediction errors in pressure heads. Subsequently, statistical measures applied to these error distributions are used to classify the occurrence and location of leaks. The suggested approach is both cost-effective and easily deployable. Through simulation-based case studies conducted on various benchmark networks, the efficacy of the proposed model is demonstrated. The results show that the model effectively predicts leak occurrences and their respective locations. However, it should be noted that as the network size increases, the model's performance diminishes, resulting in reduced accuracy. Later, the accuracy of leak prediction has been evaluated by examining its sensitivity to varying numbers of sensors and different levels of noise.

Keywords: machine learning for leak detection; leak localization; water distribution network; sensors for leak detection

Citation: Tyagi, V.; Pandey, P.; Jain, S.; Ramachandran, P. A Two-Stage Model for Data-Driven Leakage Detection and Localization in Water Distribution Networks. *Water* **2023**, *15*, 2710. https://doi.org/10.3390/w15152710

Academic Editors: Gabriele Freni and Mariacrocetta Sambito

Received: 16 June 2023
Revised: 8 July 2023
Accepted: 12 July 2023
Published: 27 July 2023

1. Introduction

Maintaining a network of pipes to supply water and remove wastewater can be challenging in an urban setting. One of the significant challenges faced by the water utilities lies in maintaining the pipe network infrastructure. Old and leaking pipes create a loss of valuable resources and give rise to health and hygiene issues. The economic angle to this problem is that the lost water fails to provide revenue to the utility. Non-revenue water (NRW) is a significant issue for water utilities in developing countries. A large portion of the water put into the distribution system is lost due to leakage and theft. Poor infrastructure maintenance leads to leakage, while inadequate metering and administrative setup lead to unauthorised consumption, resulting in NRW. For water pipelines, leakage reduces the available capacity for drinking and farming while the demand is already outstripping the supply, hurting many of the poorest people.

In Bangalore, India, Ref. [1] identified that as much as 48% of the water entering the distribution network in Bangalore is NRW. In absolute terms, around 590 million liters are lost per day. Though unauthorized access contributes to this loss, a significant part of this loss is also due to seepage and leaks in distribution mains and service pipes. Removing these leaks requires significant continuous investment and effort.

In developing urban centers, the water utility network can be vast, with heterogeneity in the pipes used and underground, making them difficult to monitor and maintain. For example, the water network in Bangalore is around 5600 km long. Parts of the piped network in the Bangalore central area may be more than 50 years old, with pipe dimensions

ranging from 50 mm to 600 mm in diameter and materials used varying from cast iron, galvanized iron, and PVC.

Typically, a hydraulic model is used to model a single-phase flow in a water pipeline network. The hydraulic parameters—the water pressure and the water flow rate at different network locations—are computed following the energy and mass conservation equations. The hydraulic model also needs to consider the water network's topological characteristics. However, there are challenges associated with developing an accurate hydraulic model for an operational urban water network. The primary challenges are (a) having accurate real-time information on the demand from different nodes of the network, (b) having sufficient information about the pipe characteristics for the entire network, and (c) a model that incorporates all the characteristics of the actual system might be challenging to build and operate.

We develop an interpretable machine-learning-based model that helps predict and localize leaks in a water network with a relatively small risk of false alarm. The proposed model uses the real-time pressure and flow measurements from a few sensors in the network to detect leaks and their locations. As the leak detection model does not require real-time demand information at each node, the exact topology of the pipeline network, and the characteristics of the pipes in the WDN, it makes the model useful for practical application. To produce a labeled dataset consisting of sensor flow and pressure readings, we need to create controlled leaks on the network's links. Consequently, the model requires some knowledge about the network, specifically the locations of the links. The model is capable of detecting leaks solely for the links it has been trained on using this labeled dataset. We consider a water network with a few select nodes where pressure and flow readings are measured and relayed in real-time using assumed sensors location. The real-time data are then fed into a two-phase algorithm. The first phase involves learning the relationship between the flow rate and the difference in pressure heads for all possible pairs of nodes with sensors in the network. In the absence of the demand information, this relationship between observed pressure head difference and flow rates at any two nodes can only partially be learned. The residual error from the predictions made by the model for a pair, and the observed difference in the pressure head of the pair, carries some information. The second phase of the algorithm then learns, in a supervised manner, the occurrence of the leak (or unaccounted demand) and its location based on the statistical changes in the distribution of the above residual errors. Here, we demonstrate the model for three diverse hypothetical benchmark water distribution networks where the data are generated using the industry-certified hydraulic model EPANET.

A significant amount of research has been conducted on detecting leaks in water distribution networks using various machine learning approaches [2–5]. The key points, including the novelty and limitations, observed using the current two-stage methodology in comparison to previous work, are discussed below:

1 The model can learn to identify leaks in all the links of the network using only a limited number of pressure and flow sensors.
2 The leak localization is not too sensitive to noise in the sensor data. Therefore the approach can provide an efficient and cost-effective solution for leak localization.
3 With only a fraction of sensors (compared to the number of links in the network), the model is capable of localizing leaks with low sensitivity to the choice of sensor location.
4 The method is also successful in identifying leaks of various sizes.
5 The two-stage methodology, unlike many other machine learning models, is interpretable, and therefore, would be better suited from an operational view for the water utilities.

As a supervised learning model, it is necessary to create labeled datasets for flow and pressure readings that correspond to leaks in each link. This can be achieved by creating controlled leaks in each link. However, generating such a dataset can be a time-consuming process. This can be seen as a limitation of the model.

The paper is structured as follows. In Section 3, we describe the hydraulic model and introduce the notations used in the paper. We present in Section 4 the simulation model used in our implementation. In Section 5, we describe the application of methodology followed for the identification of leaks. Section 6 describes numerical experiments conducted in the HANOI network ([6]), Net3 ([7]), and C town network ([8]). Finally, we conclude our findings in Section 7.

2. Literature Review

There is much research on leak detection and localization in a water distribution network using either a model-based or data-driven approach. Ref. [9] provided a support vector machine (SVM)-based approach that uses pressure data from different parts of the network to predict and localize the leak in a water network. A relatively recent line of research is focused on data-driven leak detection models. Ref. [10] provided an extensive review of the data-driven approaches for burst detection in WDN. Their recommendations include that data-driven strategies promise real-life burst detection, and reducing false alarms for such a system is an important issue. Ref. [11] used error-domain model falsification to detect leak regions in WDN. They also proposed a methodology to approximate the demand at nodes in water supply and a method for estimating uncertainties through experimentation. Ref. [12] compared the Bayesian probabilistic analysis, SVM, and artificial neural network (ANN) approach for leak detection and determined the deficiencies of these approaches under varying conditions. Ref. [13] used deep learning to narrow down the pipe burst locations from a potential district to a few pipes in a WDN. The model trained the neural network using simulated data from hydraulic models for pipe bursts. In this framework, additional pressure meters were placed at limited, optimized places for a short period (minutes to hours) to monitor system behavior after the burst and then localize the leak location. Ref. [14] used a genetic algorithm to find the size and area of a leak, such that the differences between the simulated and field-observed values for pressure head and flow were minimized.

Ref. [15] proposed a multi-stage method for leak localization within the DMA with the aid of active valve operations and smart demand metering. Each stage includes partitioning the DMA into two subregions using valve operations and identifying potentially leaking pipes within the subregions through water balance analysis based on smart demand meters. Ref. [16] used an auto-encoder neural network (AN), an unsupervised machine learning model, to detect a leak with unbalanced data—as water supply networks mainly operate under no-leak conditions. Ref. [17] used a data-driven approach based on a convolutional neural network for efficient flooding analysis and risk assessment in large urban areas.

Ref. [18] proposed a novel multiple leak detection and localization framework (MLDLF) based on the provided pressure and flow data. The methodology uses the k-means clustering method to identify leak scenarios. Ref. [19] proposed a calibration residual-based burst detection (CRBD) method that works on the output of a calibrated model and burst localization using the vector angle method. Ref. [20] proposed a pressure-data-based algorithm called the Leakage Identification and Localization Algorithm (LILA). LILA identifies potential leaks using pairwise sensor pressure data and provides the location of their nearest sensors. We refer the readers to [21] for a detailed review of recent advances in data-driven and model-based approaches for leak detection and localization.

A drawback of model-based approaches is that they require highly calibrated hydraulic models, and their accuracies are sensitive to modeling and measurement uncertainties. Hydraulic models also need real-time demand information to generate the flow characteristics for the WDN. Our proposed leak detection model does not require the topology of the network, the pipe characteristics, and real-time demand information. A data-driven model learns the relationship between the flow rates and pressure head between pairs of sensors. Changes in error (between predicted and observed readings) distributions from each sensor pair are then used as an input to the model to predict the leak and its location. Compared

to the existing models, the proposed model requires limited real-time information, is less sensitive to the placement of sensors, and can predict leak locations and occurrences with a low level of false-positive rates using a limited number of sensors.

3. Problem Formulation

We consider a water distribution network which has n_p pipes, n_j variable head nodes, and n_f fixed-head nodes. The head loss in all pipes in a network is assumed to be modeled by the Hazen–Williams formula, so the relation between the heads at two ends (node i and k) of a pipe-j and the flow is as follows:

$$H_i - H_k = r_j Q_j^n, \tag{1}$$

where Q_j is the flow in pipe p_j, H_i is the head at node i, $n = 1.852$, and r_j is the pipe resistance factor, which depends on the length, diameter, and material of the pipe. We define $\mathrm{q} = (Q_1, \ldots, Q_{n_p})^\top$ as the vector of unknown flows in the pipes.

The network topology is modeled by using the incidence matrices $\mathrm{A}_1 \in \mathbb{R}^{n_p \times n_j}$ and $\mathrm{A}_2 \in \mathbb{R}^{n_p \times n_f}$, for the unknown head nodes and the fixed head nodes, respectively. Both these incidence matrices are defined as:

$$\mathrm{A}_b = \begin{cases} -1 & \text{if the flow in pipe j enters the node i} \\ 0 & \text{if the j does not connect to the node i} \\ 1 & \text{if the flow in pipe j leaves the node i,} \end{cases} \tag{2}$$

where $b = 1, 2$. The unknown heads at different nodes are defined as $\mathrm{h} = (H_1, \ldots, H_{n_j})^\top$, the known nodal demands as $\mathrm{d}_m \in \mathbb{R}^{n_j}$, and $\mathrm{e}_l \in \mathbb{R}^{n_f}$ the fixed head elevations. Additionally we define the following matrices: O, an n_j square zero matrix, o, an $n_p \times n_j$ zero matrix, and an $\mathbb{R}^{n_p \times n_p}$ diagonal matrix G, with the following diagonal entries:

$$\mathrm{G}_{jj} = r_j |Q_j|^{n-1}.$$

The hydraulic problem is then used to solve for the unknown flow in the n_p pipes, q, and the unknown heads at the n_j nodes, h, given the network topology, A_b, the demand at nodes d_m, and fixed head elevation, e_l, such that the mass and energy for the flow are balanced. The continuity equation to be solved in matrix form can be written as (see [22] for details):

$$f(\mathrm{y}) = \begin{pmatrix} \mathrm{G} & -\mathrm{A}_1 \\ -\mathrm{A}_1^\top & \mathrm{O} \end{pmatrix} \begin{pmatrix} \mathrm{q} \\ \mathrm{h} \end{pmatrix} - \begin{pmatrix} \mathrm{A}_2 \mathrm{e}_l \\ \mathrm{d}_m \end{pmatrix} = \mathrm{o}, \tag{3}$$

where we solve for the unknown $\mathrm{y} := (\mathrm{q}^\top, \mathrm{h}^\top)^\top$. The above set of equations are typically solved using Rossman's popular program EPANET ([23]) to obtain the steady-state solution.

3.1. Simulation: Base Scenarios

We mimic an operational water distribution network, by simulating the hydraulic parameters of the network, i.e., q and h, by solving the steady-state Equation (3), for a range of fixed head elevation e_l, and nodal demands d_m. We assume that e_l, and d_m, are stochastic and drawn from a multivariate normal distribution. Specifically $\mathrm{x} := (\mathrm{e}_l^\top, \mathrm{d}_m^\top)^\top$, has the following distribution:

$$\mathrm{x} \sim \mathcal{N}(\mu, \mathbf{\Sigma}), \tag{4}$$

where μ is an $\mathbb{R}^{(n_f + n_j) \times 1}$ vector of mean head elevation and nodal demands, while $\mathbf{\Sigma}$ is the corresponding covariance matrix. If the water surface elevation level in the tanks or reservoir is constant, we set the corresponding variance entry in the covariance matrix as zero. We first draw $\mathrm{x}_1, \ldots, \mathrm{x}_{N_{base}}$, N_{base} samples of vector x, from the multivariate distribution provided in Equation (4). For a given sample of elevation and nodal demands, x_i, $i = \{1, \ldots, N_{base}\}$, the corresponding y_i^{base}, the steady-state values of the unknown heads and flows are determined by solving Equation (3). Denote P_{base} as the joint distribution of

y, obtained as above. We assume there are n_s, sensors ($n_s < n_j$), located at the subset of the n_j nodes of the network, and only the simulated pressure and flow values at these sensor nodes are used as input to our model, making it a low-dimensional representation of the high-dimensional y. The flow and pressure values at the sensor nodes, generated as above, serve as the base line data for the study.

In this work, we do not optimize the placement of sensors. Instead, we adhere to the common practice of locating the sensors at critical measuring points (CMPs). These CMPs are chosen based on the criteria that the required pressure gets lower than the minimum pressure necessary to reach the consumers at the certain nodes. Furthermore, we ensure that the sensors are positioned away from the source to account for the substantial head loss that occurs in the network as the supply reaches the farthest points.

3.2. Simulation: Scenarios with a Leak

Following [24], the demand due to leak—the mass flow rate of fluid through the hole—is expressed in a general form:

$$d_{leak} = C_d A p^{\alpha} \sqrt{\frac{2}{\rho}}, \tag{5}$$

where d_{leak} is the equivalent water demand due to leak, C_d is the discharge coefficient, where we use a default value of 0.75, A is the area of the hole in the pipe, p is the internal water pressure, ρ is the density of the fluid, and the exponent α is a unitless parameter related to the characteristics of the leak. We use the default value of $\alpha = 0.5$, which results in an equivalent equation:

$$d_{leak} = C_d A \sqrt{2gh},$$

where g is the acceleration of gravity and h is the gauge head. We use the Water Network Tool for Resilience (WNTR) ([25]), a Python-based package that is compatible with EPANET, for our simulation for both leak and no leak scenarios. The leak is added to a location in a pipe by splitting the pipe into two sections and adding a node with the demand characteristics, as specified in Equation (5).

With this additional leak-node k, and N^k_{leak},k with stochastic demands $\mathrm{x}_1, \ldots, \mathrm{x}_{N^k_{leak}}$, sampled from the distribution in Equation (4), the corresponding y^{leak}, the steady-state values of the unknown heads and flows, for all the nodes and pipes of the network are then obtained by solving Equation (3). These samples represent flow characteristics with an incremental leak added to the network. We denote P^k_{leak} as the joint distribution of y, obtained as above.

Given a series of pressure and head values at the n_s sensor nodes, we want to label them as data from the base scenario y^{base}, or the leak scenario y^{leak}. In case of leak scenarios, we would like to identify the node k where the leak was introduced.

4. Methodology

Let S be the set of all nodes in the network. Define S_s, a subset of S with cardinality n_s, containing any n_s elements in S. S_s can be seen as the set of nodes where the sensors for measuring pressure heads and flows are located. Define $\mathrm{h_s} \equiv (H_i)^\top, i \in S_s$ and $\mathrm{q_s} \equiv (Q_i)^\top, i \in S_s$ as a vector of pressure heads, and flows measured, respectively, at the nodes with the sensors for the base network (where no additional leak demand node is added). We first model the relationship between the nodal pairs in S_s, for the base network. There will be ${}^sC_2$ such pairs, and for each pair, we fit a linear regression model:

$$\mathbb{E}_{\mathrm{y}\sim P_{base}}\left[\Delta h_{ij} | Q_i, Q_j\right] = \beta_0 + \beta_1 Q_i + \beta_2 Q_i^2 + \beta_3 Q_j + \beta_4 Q_j^2, \tag{6}$$

where $i, j \in S_s$, $i \neq j$, $\Delta h_{ij} = H_i - H_j$, and ϵ_{ij} is the unexplained error for the pair. The residual error for the $i-j$th pair is defined as:

$$\epsilon_{ij} = \mathbb{E}_{y \sim P_{base}}\left[\Delta h_{ij} | Q_i, Q_j\right] - \Delta h_{ij}. \tag{7}$$

Of The N_{base}, base scenarios generated, as described in Section 3.1, a subset $N_{base}^{Training1}$ is used to obtain the coefficients for the $i-j$th pair, $\beta_{ij} \equiv (\beta_0, \ldots, \beta_4)^\top$, using the ordinary least squares regression. The flow readings for the pair can only partially explain the variation in the difference between pressure heads for the two nodes, as we cannot fully explain the outcomes of the high-dimensional model (as described in Section 3) using a low-dimensional dataset. If the high-dimensional datasets are available, a machine learning model could precisely learn the functional relationship between flow and pressure heads. Any deviation from this learned relationship would be adequate for leak detection. In the current setup, as only partial information of this high-dimensional data is available, the model will make predictions with error. However, changes in the distribution of the prediction errors can be used to localize leaks. This is what is achieved in the second stage of the model. However, as long as the network topology, and the distribution of the input to the hydraulic model is fixed, i.e., $x \sim \mathcal{N}(\mu, \mathbf{\Sigma})$, the distribution of y and residual error ϵ_{ij} will remain unchanged. As ordinary least squares is an unbiased estimator, the mean of ϵ_{ij} would be close to zero [26].

4.1. Identifying the Occurrence of Leaks

Once an additional leak node is added to the network, the distribution of the outcomes, y, changes from P_{base} to P_{leak}. Now, Δh_{ij} and Q_i, Q_j are not drawn from the distribution P_{base}, used to train the model in Equation (6), and thus, the distribution of the residuals, as obtained in Equation (7) cannot be guaranteed to be centered around mean zero, or in general have the same distribution as when Δh_{ij} and Q_i, Q_j, are drawn from P_{base}. As we are dealing with a discrete sample of y, and as a result discrete samples of ϵ_{ij}, if we have known sampled errors for the base case (which serves as the reference empirical distribution), we can infer the change in the distribution of y, by statistically measuring the changes in the distribution of ϵ_{ij}. The Kolmogorov–Smirnov test (KS test) is a commonly used statistical non-parametric test of the equality of one-dimensional probability distributions that can be used to compare two samples. If $F_{1,n}^{ij}$ is the empirical distribution function for the base residual errors with n iid samples for nodal pair $i-j$, and $F_{2,m}^{ij}$ is the empirical distribution function for unknown residual error (it is unknown whether it is an error when base scenarios or scenarios with a leak at k are used) with m iid samples for the same pair, then the KS-statistics is defined as:

$$D_{n,m}^{ij} = \sup_x |F_{1,n}^{ij}(x) - F_{2,m}^{ij}(x)|. \tag{8}$$

We could use the KS statistics, $D_{n,m}^{ij}$, to accept or reject the null hypothesis that sample 2 is drawn from the same distribution as sample 1 for a given level α, and therefore, draw conclusions with a certain level of confidence—whether there is a leak in the network or not—however, it would not help us deduce at which node the leak exists. This is because there could be several nodal pairs for which the distribution of residual errors could change when a leak is introduced in the network.

4.2. Identifying the Location of Leaks

We consider a case where a single incremental leak is introduced to the network at one of the n_p links. We want to develop a supervised learning-based model that can, given a sequence of pressure head and flow readings at the nodes with sensors, help us infer which of the n_p links the leak was introduced. We use a multinomial logistic regression to model the posterior probabilities of the occurrence of a certain class (here, a leak at one of

the n_p links or no leak). For our leak detection model, we use $z = (D^{ij})^\top \, i, j \in S_s, i \neq j$ as the predictors. As the residuals are normally distributed, alternatively, we also consider the following statistics for our predictors:

$$\mu_1^{ij} = \frac{1}{n} \sum_{l=1}^{n} \epsilon_{ij}^{base},$$

and

$$\mu_2^{ij} = \frac{1}{m} \sum_{l=1}^{m} \epsilon_{ij}^{unknown},$$

namely, $z = (\mu_1^{ij}, \mu_2^{ij})^\top \, i, j \in S_s, i \neq j$. Here, n and m are the number of samples we pick from the base scenarios and scenarios with a particular label (leak at one of the nodes among K or no leak). While training, the label k would be known, and we would then test on a new test dataset, where the label is not revealed to test the accuracy of the model. We generate $z_1, \ldots, z_{N_k}^{train2}$, N_k^{train2}, samples for each label $k = 0, \ldots, n_p$, with $k = 0$ being the label for no leak, $k = 1$ corresponds to a leak at link 1, and so on, and δ is the coefficient.

The logistic regression model would then be of the form:

$$\begin{aligned} Pr(G = k|Z = z_i) &= \frac{\exp(\delta_{k0} + \delta_k^\top z_i)}{1 + \sum_{l=0}^{K-1} \exp(\delta_{l0} + \delta_l^\top z_i)}, k = 0, \ldots, n_p, \\ Pr(G = K|Z = z_i) &= \frac{1}{1 + \sum_{l=0}^{K-1} \exp(\delta_{l0} + \delta_l^\top z_i)} \end{aligned} \tag{9}$$

4.3. Summary of the Method

Here, a summary of the steps involved is described.

Generating base scenarios and regression:

Step 1: Draw N_{base} demands from the multivariate normal distribution.

Step 2: Using each set of the N_{base} demands, solve for flow and pressure at the links and nodes of the network.

Step 3: Retain the flow and pressure data for nodes marked as sensors and mark the dataset as base scenarios.

Step 4: For each sensor node pair, perform an ordinary least squares using a subset of the N_{base} base scenarios, following Equation (6).

Generating labeled leak scenarios:

Step 5: Introduce leak node with a given area at the center of the jth link, $j \in [1, n_p]$.

Step 6: For the demand in the remaining nodes, draw N_{leak} samples from the multivariate normal distribution.

Step 7: Using each set of the N_{leak} demands, with an additional leak node as described in Step 5, solve for flow and pressure for the network.

Step 8: Retain the flow and pressure data for nodes marked as sensors and mark the N_{leak} dataset with a leak link id j.

Step 9: Repeat Step 5 to Step 8 for all the links.

Generating predictors:

Step 10: From the N_{leak} scenarios labeled as 'leak in link j', pick $m \leq N_{\text{leak}}$ randomly chosen scenarios without replacement.

Step 11: For each of the m scenarios, find the residual errors for each sensor pair, where prediction is made using coefficients from [Step 4].

Step 12: Pick $n \leq N_{\text{base}}$ randomly chosen scenarios without replacement from the N_{base} base scenarios.

Step 13: For each of these n scenarios, find the residual errors for each sensor pair, where a prediction is made using coefficients from [Step 4].

Step 14: Compute the predictors for each sensor pair $i-j$, i.e., D^{ij}, μ_1^{ij}, and μ_2^{ij}, $i, j \in S_s, i \neq j$.

Step 15: Repeat Step 10 until Step 14 $N^{train2}_{\text{leak in link j}}$ times, to obtain $z_1, \ldots, z^{train2}_{N_{\text{leak in link j}}}$, with label 'leak in link j'.

Step 16: Repeat Step 10 till Step 15 for a leak in all $j \in [1, n_p]$ links.

Classification:

Step 17: With labeled z data, train the logistic regression model, as described in Section 4.2 with appropriate thresholds for multinomial classification of leak location. The process flow of approach is as shown in Figure 1.

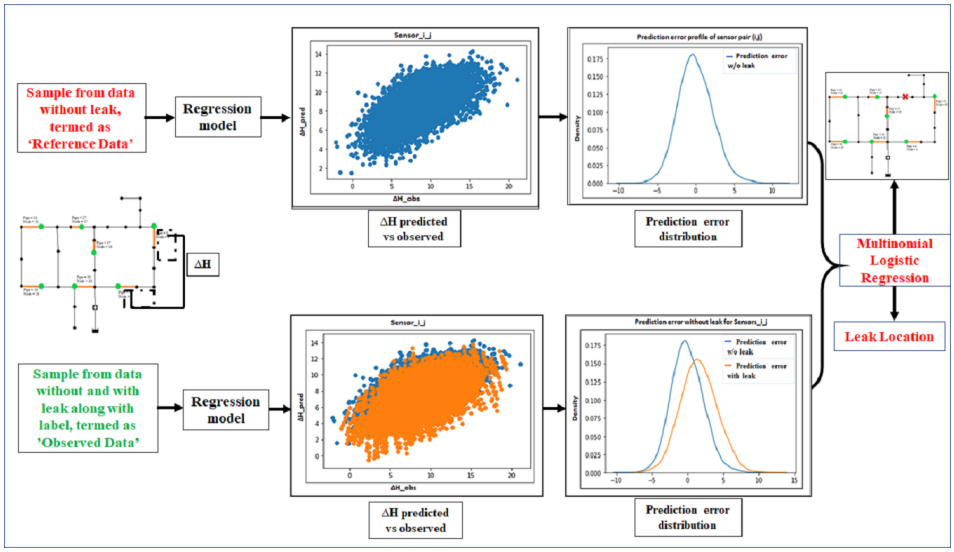


Figure 1. Process flow of linear regression and logistic regression.

5. Application of the Methodology on Water Distribution Networks

5.1. Data Simulation

The base water demand in all the nodes of the network is assumed to be known. Several artificial demand patterns were generated assuming a multivariate Gaussian distribution for the joint demand distribution at all the nodes. The mean of the multivariate Gaussian distribution is taken as the base water demand. The demand standard deviation was assumed to be in between 16% and 20% for all the nodes (we aimed to examine a scenario involving high variability in the demand at nodes, which is often the case as temporal demand values fluctuate throughout the day. Additionally, we conducted a sensitivity analysis regarding the demand variability and found that the model's outcome is not significantly influenced by this variability). A constant correlation coefficient of 0.6 between any two pairs of nodes was used to generate the covariance matrix. The above choice of standard deviation allows wide variation in demand, which is typically expected throughout the day. EPANET 2.0 and WNTR Python package were used to perform hydraulic simulation of this network for the various demand scenarios generated above (the python code and scripts used for the experiments are made available at https://github.com/adOption/DPTrans (20 September 2021)). The node heads and link flow values from the simulation at specified pre-defined nodes were used instead of the sensor data.

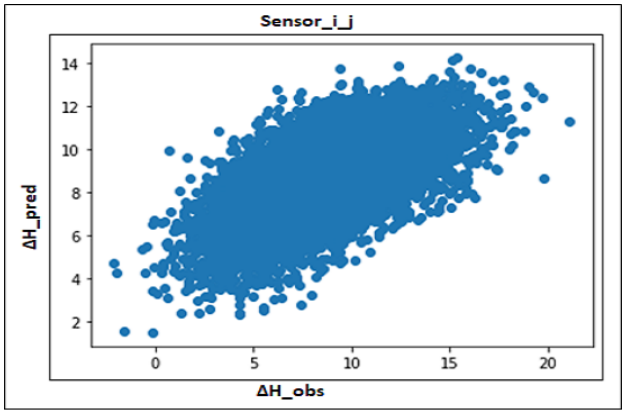
A leak node was introduced in each of the n_p links, one after the other. These leaks were positioned exactly at the midpoint of each link for training the model while for creating the test dataset, the leaks were introduced randomly at either 3/10 and 7/10 of the

length of the pipe. For the training dataset, we create equal samples of the leak node with four variations of leak area, i.e., 0.0005 m^2, 0.002 m^2, 0.003 m^2, and 0.004 m^2. For the test dataset, the area had the following three variations, 0.0001 m^2, 0.001 m^2, and 0.005 m^2. The generated scenarios were labeled as $k = 0$ for no leak, $k = 1, \ldots, n_p$, where n_p is equal to the number of links. The flow in the links and pressure head at sensor nodes were recorded along with the labels for both training and test datasets.

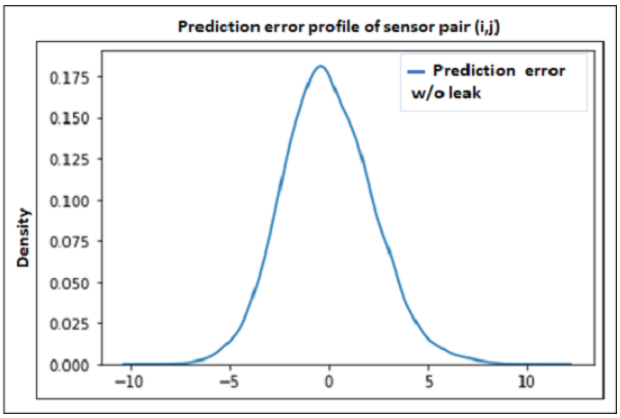
The experimental configuration for all the case studies is illustrated in Figures mentioned in section 6 . The nodes where pressure sensor is introduced is illustrated by green circles, and the links where flow sensors are placed is depicted by orange lines. The choice of these sensor locations is based on factors such as criticality and distance from the sources. However, the sensor placement optimization becomes less critical when sufficient number of pressure and flow sensors are available, as would be demonstrated in the case studies discussed in Section 6. In the model setup, only one leak node is active at any given time.

5.2. Regression

We fit regression models for each sensor pair combination using a subset of samples from the base no-leak scenarios (there will be ${}^nC_2$ such sensor pairs). For any given nodal pair $i - j$, the regression model predicts the Δh_{ij}, given the flow rates at the links associated with the two nodes. As discussed earlier, the regression model will only be able to partially explain the delta variations in the head. Figure 2 shows the predicted $\Delta h_{i,j}$ from the linear regression against the observed $\Delta h_{i,j}$ for the sensor node pair i and j, when only data from the base scenarios are used. Furthermore, the residual error is plotted, which, as expected, is centered around a mean of 0 and has a standard deviation lower than the standard deviation of the $\Delta h_{i,j}$ observed.



(**a**) $\Delta h_{i,j}$ observed vs. predicted.



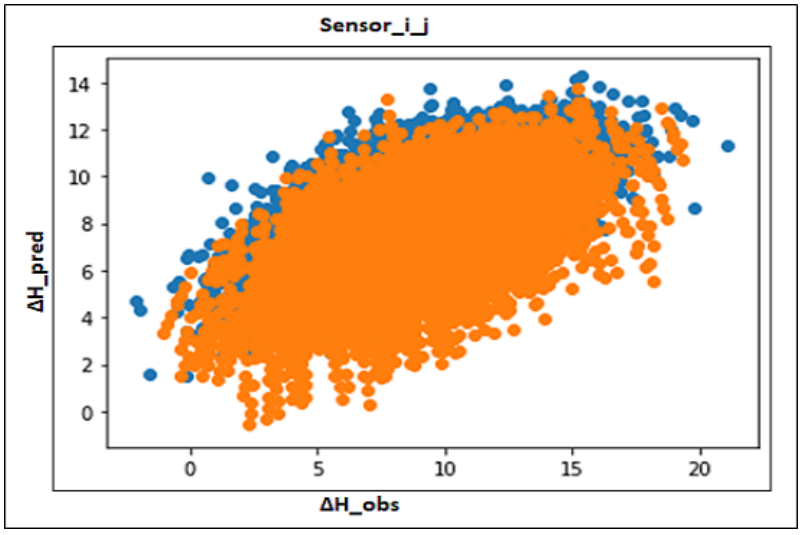
(**b**) Distribution of $\epsilon_{i,j}$ observed

Figure 2. For the sensor node pair i and j, the regression model helps in partially explaining the variations observed in $\Delta h_{i,j}$.

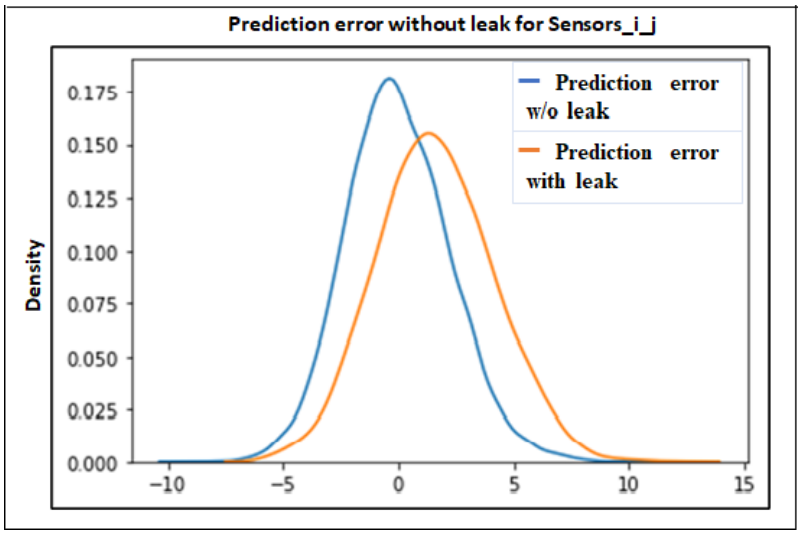
5.3. Impact of Leaks on Residual Errors

Using the regressed functions for the sensor pairs, we predict Δh_{ij} using flow data from scenarios that we labeled as no leak when $k = 0$ and leak nodes when $k = 1, \ldots, n$, where n = number of links. The residual errors between the predicted Δ_h and observed Δ_h for scenarios with leaks differ in distribution when compared to the corresponding residual error distribution when only base scenarios are used.

Figure 3 compares the residual error in prediction of $\Delta h_{i,j}$ and $\Delta h_{i,j}$ for no-leak scenarios and scenarios with a leak at any link in the network. While the distribution of residual errors is unbiased for the base scenarios, the errors are no longer guaranteed to be unbiased when a leak is introduced. Additionally, a single leak will impact the error distribution between most sensor nodal pairs. In Figure 3, we see that a leak introduced in any link impacts both the situation when there is no leak and when there is a leak.



(**a**) $\Delta h_{i,j}$ observed vs. predicted.



(**b**) Distribution of $\epsilon_{i,j}$ of no leak and leak case

Figure 3. For the sensor node pair i and j, regression model helps in partially explaining the observed variations in $\Delta h_{i,j}$ when a leak is introduced in a link.

5.4. Classification

Figure 3 demonstrates that the introduction of a leak in the network would impact the residual errors across sensor nodal pairs. Even while comparing the distribution of

residual errors between base no-leak and labeled no-leak scenarios, there will be deviations depending upon the sample size n, m used. As discussed in Section 4.2, we, therefore, use the supervised learning model of multinomial logistic regression to classify whether a set of residual errors are coming from $k = 0$, i.e., the no-leak case, or with a leak in the kth link, when $k \in [1, \ldots, n_p]$. The input to the logistic regression model is the vector z, whose elements are as described in Section 4.2. We use a set of $n = m = 500$, error samples from the base no-leak scenarios and scenarios labeled as a leak at a particular link, respectively. For this set, as described in Section 4.1, we obtain the K-S statistics, the sample mean for the errors with base no-leak scenarios, and the sample means of errors with the labeled scenarios to obtain a single instance of z.

We test the efficacy of the trained model by predicting the occurrence of leak and location of the leak node for a test dataset with $N_{k=0}^{test} = 500$, for the no-leak case, and $N_k^{test} = 500$, $k = 1, \ldots, n_p$ for the different links with a leak. The predictors for the test data are prepared following the same procedure as that for the training data. The error metrics to assess the performance of the leak classification has been generated using the test dataset.

5.5. Impact of Noisy Sensor Data

To make the simulated experiments closer to reality, we want to account that the sensor data typically would have an additional measurement noise. Thus, we add Gaussian white noise to the simulated flow and pressure data at the sensor nodes. We varied the standard deviation of the white noise used between 0.25% and 10%. Thus, we have,

$$\widehat{\mathrm{h}} = \mathrm{h}(1 + \mathrm{e}_h^\top) \quad \mathrm{e}_h \sim \mathcal{N}(o, \Sigma_e);$$

where Σ_e is a diagonal covariance matrix with diagonal entries as σ for the appropriate noise level. Similarly:

$$\widehat{\mathrm{q}} = \mathrm{q}(1 + \mathrm{e}_q^\top) \quad \mathrm{e}_q \sim \mathcal{N}(o, \Sigma_q).$$

We use the noisy pressure head data and flow data as the inputs to our model.

5.6. Error in Terms of Topological Distance

Topological distance refers to the shortest distance measurement between two links. In this study, it is utilized to determine the distance between the true and predicted leak locations. It provides an indication of how far the model has predicted the leak location in comparison to the original location by counting the number of links between the two along the shortest path. Alternatively, instead of counting the number of links between the true and predicted leak location, one can also directly measure the distance between them along the shortest path. The python package NetworkX has been used to measure the topological distance [27].

We introduce the term average topological distance (ATD), as the average of the topological distance between the predicted and true leak location for the N_{test} scenarios. The ATD should be close to zero for a good leak localization model.

6. Results

The leak detection framework described in the previous sections is applied to the three different WDNs to demonstrate its effectiveness. These are composed of:

(a) Hanoi WDN (WDN1);
(b) Net3 (WDN2);
(c) C-Town Network (WDN3).

As shown in Figure 4. The example networks considered vary from each other in terms of their size and complexity. The WDN1 consists of $n_j = 31$ nodes and $n_p = 34$ pipes. The WDN2 has $n_j = 92$ nodes, $n_p = 117$ pipes, 2 reservoirs, and 3 tanks, while WDN3 has $n_j = 388$ nodes, $n_p = 429$ pipes, 1 reservoir, and 7 tanks. The other topological details of the networks can be found in [6–8], for WDN1, WDN2, and WDN3, respectively.

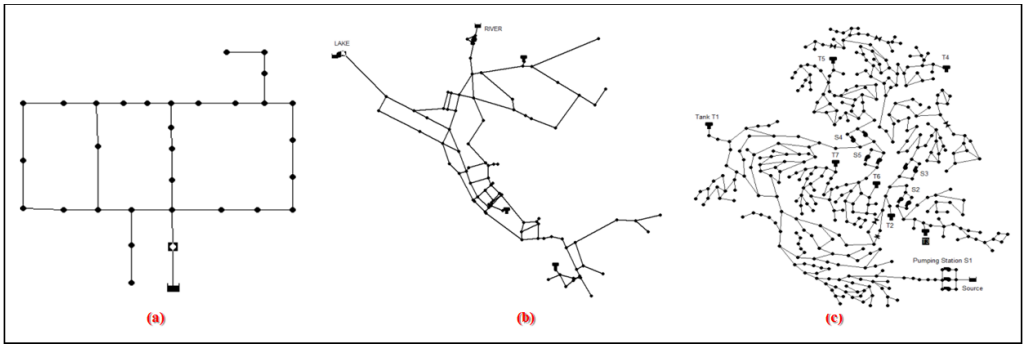


Figure 4. Network topologies of the three example networks, (**a**) Hanoi WDN, (**b**) Net3 WDN, and (**c**) C-town WDN.

We use a combination of critical measuring points (CMP) and distance from the source for identifying the sensor location. The choice of location for the sensors in the three networks is provided in Figure 5. The pressure and flow sensors are assumed to be positioned in the same location. This assumption is necessary because the regression in Stage 1 focuses on modeling the relationship between flow and pressure using a pair of sensors.

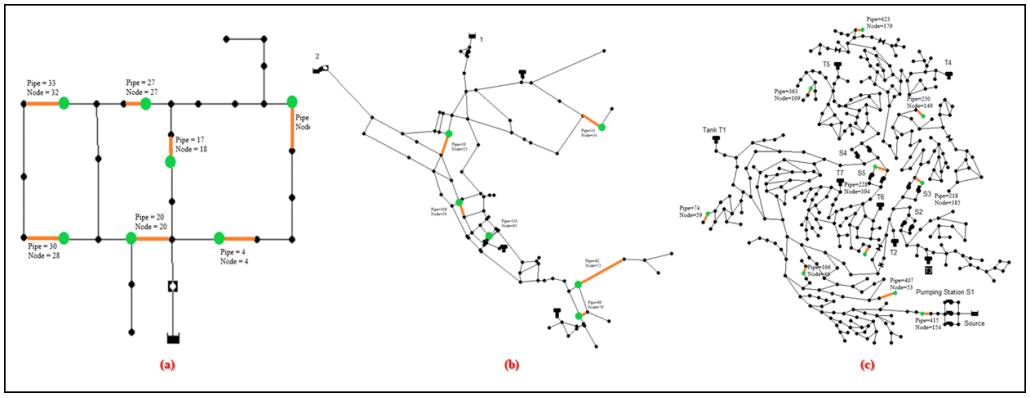


Figure 5. Network topologies with sensors location of the three example networks (**a**–**c**). The green dot represents the location of the pressure sensor, while the orange line represents the presence of the flow sensor.

6.1. Case Study 1: Hanoi WDN (WDN1)

The developed methodology is initially implemented and demonstrated on a small-scale water distribution network. The sensors are strategically installed at seven locations in links 4, 18, 17, 20, 27, 30, and 33 to capture pressure and flow data. Linear regression models are then fitted to predict the measured pressure difference between each sensor pair using the flow readings for the pair. Subsequently, a multinomial logistic regression is fitted to classify the leak location for the labeled training dataset. Finally, the model is used on a separate test dataset to predict the location of the leak in the 34 links of the network. The model is evaluated for varying number of sensors used and different levels of noise in the sensor reading.

Experiment with Varying Number of Sensors and Noise in Sensor Data

We first test the accuracy in predicting the leak location (including the identification of the no-leak cases) for the network using three sensors (links 8, 20, and 27), five sensors (links 4, 18, 17, 30, and 32), and seven sensors (links 4, 18, 17, 20, 27, 30, and 33) and for the following noise levels: 0.5%, 1%, 2.5%, 5%, and 10%. The accuracy is defined as the number of scenarios where the link with a leak was correctly identified over the total number of scenarios. We see in Figure 6 that the best accuracy in identifying leak location is 91%, while the lowest accuracy achieved was 79%. The accuracy increases with number of sensors

used, although roughly similar levels are achieved with five and seven sensors. As the noise in the sensor data increases, as expected, we see poor performance in identifying the leak location accurately. However, even when 2.5% noise was added to the sensor data, the method achieved satisfactory accuracy in leak location identification.

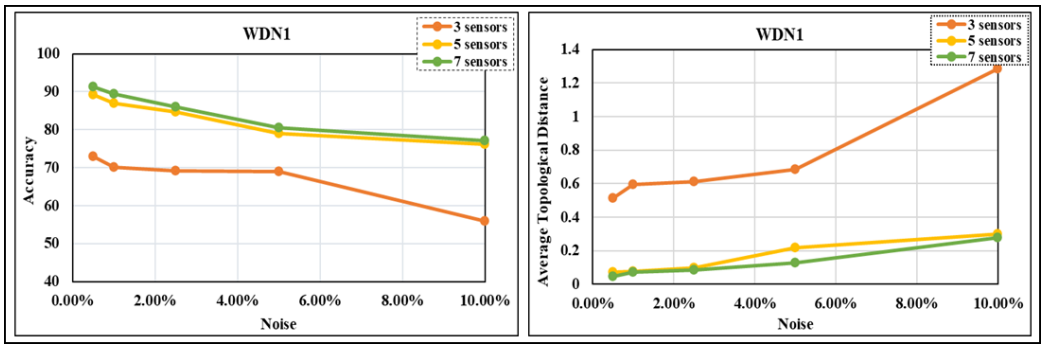


Figure 6. Accuracy and ATD variation with Noise Levels and Sensor Count in WDN1.

Another measure that we use to check the efficacy of the model is the ATD. The ATD measures, on average, how far from the actual leak link the algorithm predicted the leak location (Section 5.6). A good leak detection algorithm would have an ATD value close to zero. We see in Figure 6 that ATD increases as we reduce the number of sensors, or increase the level of noise in the sensor reading. We see that even for the worst case, with only three sensors and a noise of 10% in the sensor reading, the algorithm on average predicts the leak link within 1.3 links from the actual leak location. Notably, ATD achieved using five sensors is as good as results obtained using seven sensors.

6.2. Case Study 2: Net3 (WDN2)

To evaluate the scalability of the model described in the previous case study, its application is expanded to a medium-sized network. The model is trained to detect leaks in any of the 117 links within this network, which is relatively more complex than WDN1. Unlike WDN1, WDN2 comprises five sources (three tanks, two reservoirs) and two pumping stations (assumed as non-operational). For this network, a total of six sensors are strategically installed at links 18, 31, 82, 89, 108, and 110 to record flow and pressure data. Additionally, the model is tested with varying numbers of sensors and different levels of noise in the sensor data.

6.2.1. Experiment with Varying Number of Sensors and Noise in Sensor Data

The accuracy in predicting the leak location (including the identification of the no-leak cases) is tested for three sensors (links 31, 89 and 110), five sensors (links 18, 31, 82, 89 and 108) and six sensors (links 18, 31, 82, 89, 108, and 110) and for noise levels of 0.5%, 1%, 2.5%, 5%, and 10%. From Figure 7, we observed that the highest accuracy achieved in identifying leak locations is 79%, while the lowest accuracy obtained is 49%. The accuracy improves as the number of sensors used increases, although similar levels of accuracy are roughly attained with five and six sensors. As the level of noise in the sensor data increases, a decline in the accurate identification of leak locations is expected. However, even when 1% noise was introduced to the sensor data, the model was still able to achieve satisfactory accuracy in identifying leak locations.

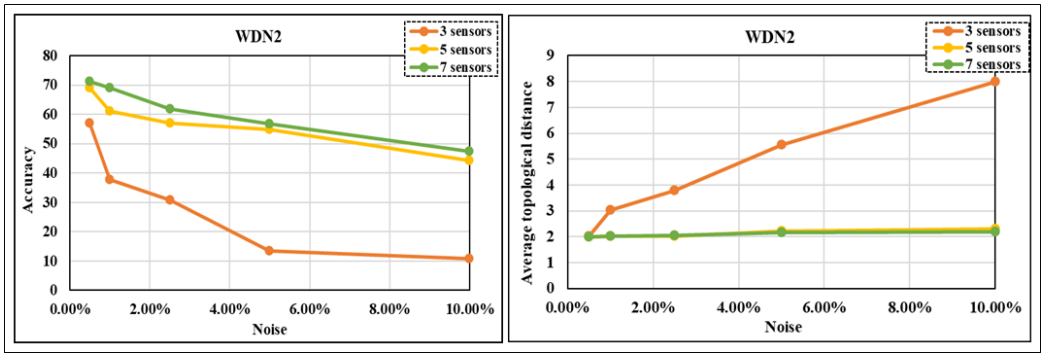


Figure 7. Accuracy and ATD variation with noise levels and sensor count in WDN2.

We can observe from Figure 7 that as the number of sensors decreases or the level of noise in the sensor readings increases, the ATD increases. The ATD in the best scenarios (with five and six sensors) ranges between 2 and 2.3. Even in one of the worst-case scenarios, with only three sensors and a 2.5% noise level in the sensor readings, the algorithm achieves an average prediction of the leak link within a distance of 3.8 links from the actual leak location. Importantly, the ATD attained using five sensors is comparable to the results obtained with six sensors, highlighting the effectiveness of the algorithm with a reduced sensor count.

6.3. Case Study 3: C-Town Network (WDN3)

The framework is deployed in a larger benchmark network with increased complexity, consisting of one reservoir, seven tanks (eight sources), and three valves (considered fully opened). Two test cases are conducted, involving different numbers of sensors (10 and 5). The initial placement of sensors is performed at links 423, 250, 363, 228, 218, 74, 166, 407, 415, and 136, which correspond to 10 specific links. The accuracy and ATD are then compared across five distinct levels of noise. Additionally, the model is employed to predict the location of the leak in the 429 locations of the network. The model is evaluated for the varying number of sensors used and different levels of noise in the sensor reading.

6.3.1. Experiment with Varying Numbers of Sensors and Noise in Sensor Data

The accuracy in predicting leak locations (including the identification of non-leak cases) is tested using five sensors (links 166, 72, 228, 250, and 423) and another with ten sensors (links 423, 250, 363, 228, 218, 74, 166, 407, 415, and 136) and for the noise levels as predefined in Section 6.2.1. The highest accuracy achieved when locating leaks is 30%, while lowest accuracy observed is 10%, as shown in Figure 8. The accuracy improved with an increase in the number of sensors. Additionally, accurate identification of leak locations is observed as the noise levels in the sensor data increase. However, even with minimal noise levels, the accuracy was affected, resulting in a reduction to 25%.

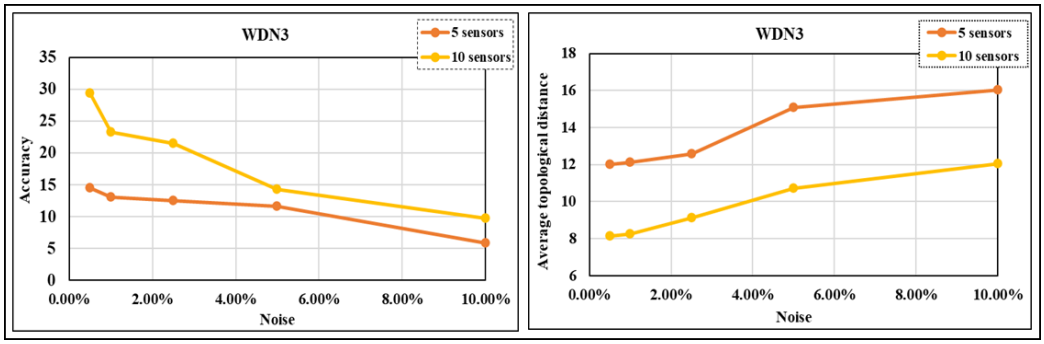


Figure 8. Accuracy and ATD variation with noise levels and sensor count in WDN3.

From Figure 8, it is observed that as the number of sensors decreases or the level of noise in the sensor readings increases, there is an increase in ATD. In the best-case scenarios (with ten sensors), the ATD ranges from 8 to 12. Even in one of the worst-case scenarios, where only five sensors are used and the sensor readings have a noise level of 0.5%, the algorithm achieves an average prediction of the leak location within a distance of 12 links from the actual leak location.

6.3.2. Comparative Analysis of the Results from Three Networks

The results obtained using the framework is compared, as shown in Figure 9.

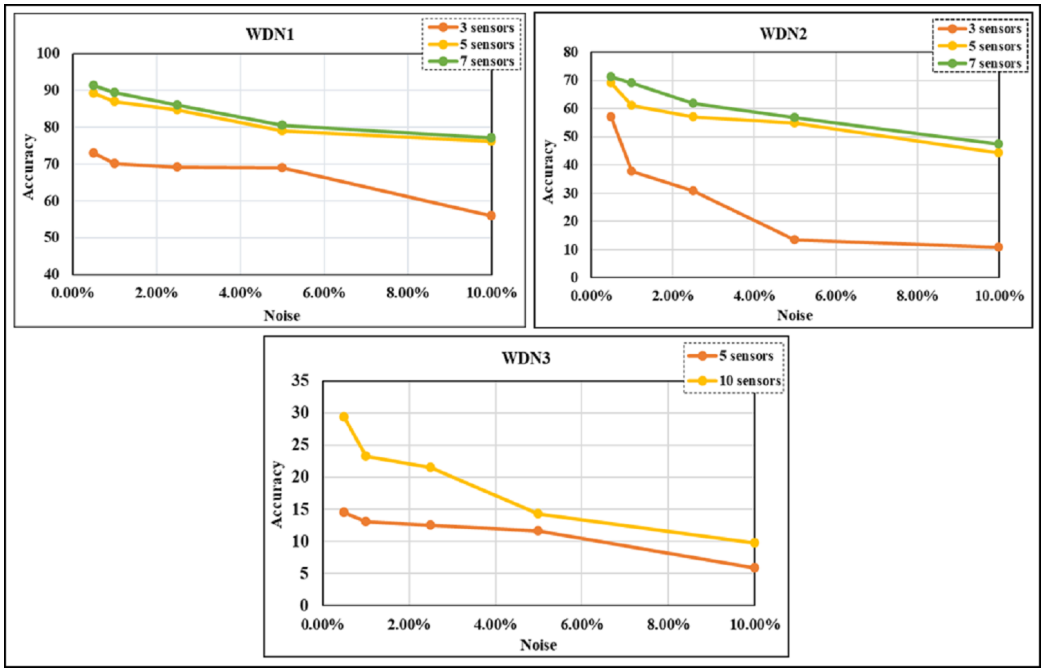


Figure 9. Comparison of the accuracy achieved across three distinct water networks.

It is evident from the figure that as the size and complexity of the network increases, the obtained accuracy proportionately decreases. The highest accuracy obtained for the three networks WDN1, WDN2, and WDN3 is 92%, 79%, and 34% respectively.

While the accuracy obtained in identifying the leak location in larger network is low, a closer look at the predicted leak location shows that the method still provides useful insights. Figure 10 illustrates a few examples of true leak location and the corresponding predicted link with a leak. For our accuracy measurement we only count those scenarios where the link with a leak is precisely identified as correct classification.

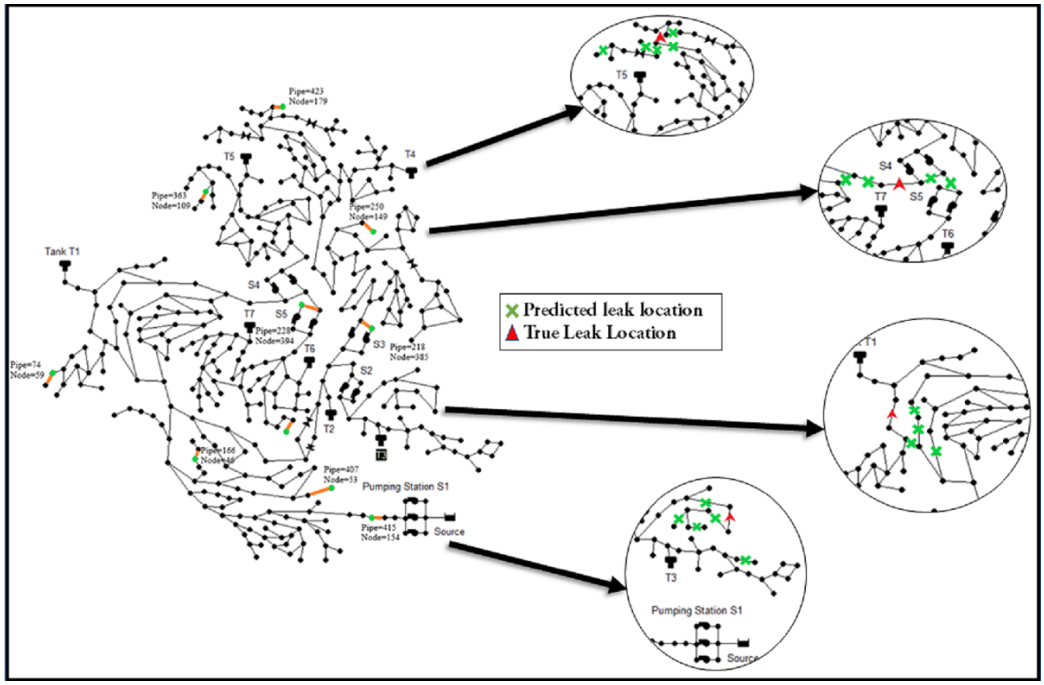


Figure 10. Comparison between actual and predicted leak locations in WDN3.

If we provide a threshold instead, where the predicted link with a leak is within the given tolerance of topological distance, we classify it as a correct identification.

In order to evaluate the accuracy of the model in classifying the leak location more comprehensively, we classify cases where the predicted link with a leak is within a topological distance of 2, 5, 8, 10, and 20 links from the true leak location. Figure 11 shows that even for a significant level of noise in the sensor reading, using ten sensors, the algorithm is able to predict leaks within a topological distance of 10 links. A practical use case is that using a few sensors the leak location is coarsely identified in the network and the actual location can then be identified by additional instrumentation around this region.

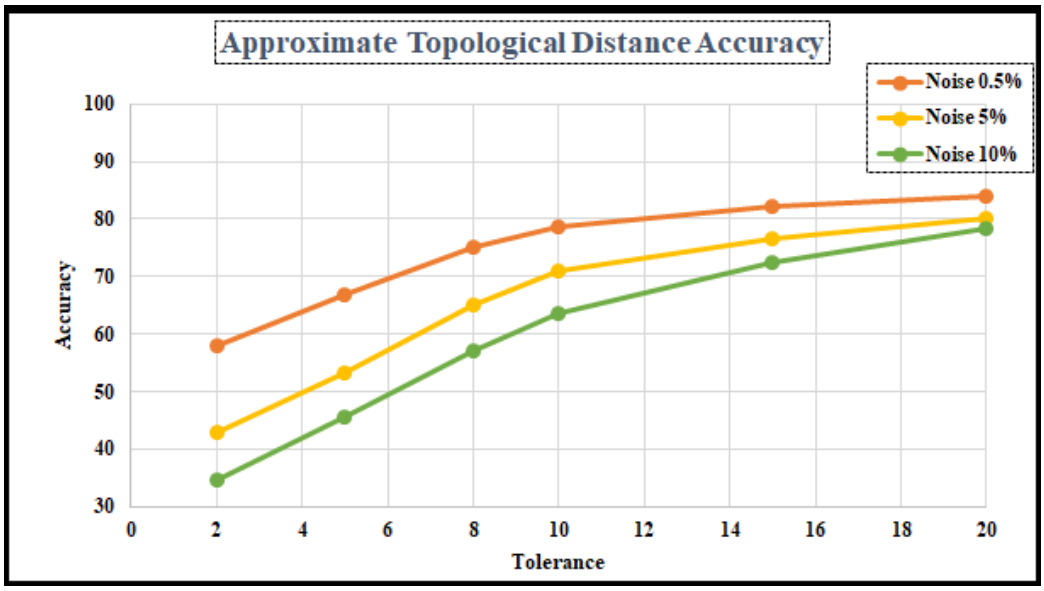


Figure 11. Accuracy of the leak location classification of the WDN3, corresponding to different levels of tolerance in topological distance between the true and the predicted link with a leak.

7. Conclusions

We have developed an algorithm to detect leaks in water distribution network using a two-stage methodology, where in the first stage we learn the relationship between the flow and pressure readings of the installed sensors in the network with no leaks. In the second stage, using a labeled dataset of leak location, we fit a multinomial logistic regression model using as input the change in the error distribution of the predicted head between sensor pairs. The premise of our approach is based on the observation that the introduction of a leak alters the distributional properties of hydraulic parameters, specifically pressure and flow rates. By measuring the flow characteristics at specific locations in the network, we aim to distinguish between different leak locations or the absence of a leak by mapping changes in a high-dimensional distribution to a low-dimensional error distribution. Introduction of a leak will result in a discernible change in the error distribution, enabling us to locate the leaks effectively.

By utilizing a simulated dataset, we implemented our approach on three benchmark networks, namely WDN1, WDN2, and WDN3. The results demonstrated that our model exhibits reasonably good accuracy in classifying leak locations within these networks. Specifically, for WDN1 and WDN2, we observed that our approach successfully identified leaks even when they had a small orifice area. However, as the noise in the data increased and the number of sensors decreased, the method occasionally resulted in misclassifcation. Nevertheless, it is worth noting that even when the number of sensors was slightly reduced, the model maintained a relatively high level of accuracy.

In contrast, when dealing with a larger-sized water network WDN3, the accuracy of our approach was significantly compromised. It became evident that the accuracy did not hold up well as the network size increased. We observed a substantial number of misclassifications in accurately identifying the specific links where leaks occurred. However, as the predicted leak location links where in close neighbourhood of the true leak location, we incorporated a tolerance factor in classifying whether the leak location has been correctly identified. This means that even if the identified leak location was in adjacent links rather than the exact one, it would still be considered an acceptable classification. With this modification, the accuracy of identifying leak in the network was found to be close to 60%, even in the presence of high levels of noise in the sensor readings. Therefore, while the method cannot precisely identify the leak location, we show that it is able to identify the area within which the leak is present.

In summary, our algorithm exhibited promising performance in classifying leak locations in the benchmark networks, demonstrating its potential usefulness in practical scenarios. Despite some challenges posed by increased noise and reduced sensor numbers, the model's accuracy remained satisfactory, showcasing its robustness to certain variations in the data. By creating a labeled dataset of controlled leaks in the links of a real WDN, the presented approach can be used to learn to locate new leaks in the network. As a future step, the two-stage methodology will be applied to a real-life network, which will help validate the assumptions of the simulation-based model.

Author Contributions: V.T. played a significant role in shaping the methodology, conducting formal analysis, and validating the results. P.P. was primarily responsible for crafting the initial draft and overseeing its revision. S.J. played a key role in conceptualizing the methodology and validating the results. Furthermore, he made remarkable contributions in writing, reviewing, and editing the main draft. P.R. provided invaluable technical expertise, resource coordination, problem formulation, and meticulous review and editing of the final draft. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No data were utilized in this study. Therefore, data availability is not applicable.

Acknowledgments: The authors would like to thank MeITY, India, for providing financial support to implement the "DP-Trans (Digital twin for Pipeline TRANSport Network) project".

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

References

1. Raj, K. Sustainable urban habitats and urban water supply: Accounting for unaccounted for water in Bangalore City, India. *Curr. Urban Stud.* **2013**, *1*, 156. [CrossRef]
2. Mashhadi, N.; Shahrour, I.; Attoue, N.; El Khattabi, J.; Aljer, A. Use of machine learning for leak detection and localization in water distribution systems. *Smart Cities* **2021**, *4*, 1293–1315. [CrossRef]
3. Fan, X.; Yu, X. An innovative machine learning based framework for water distribution network leakage detection and localization. *Struct. Health Monit.* **2022**, *21*, 1626–1644. [CrossRef]
4. Fares, A.; Tijani, I.; Rui, Z.; Zayed, T. Leak detection in real water distribution networks based on acoustic emission and machine learning. *Environ. Technol.* **2022**, 1–17. [CrossRef]
5. Kammoun, M.; Kammoun, A.; Abid, M. Experiments based comparative evaluations of machine learning techniques for leak detection in water distribution systems. *Water Supply* **2022**, *22*, 628–642. [CrossRef]
6. Fujiwara, O.; Khang, D.B. A two-phase decomposition method for optimal design of looped water distribution networks. *Water Resour. Res.* **1990**, *26*, 539–549. [CrossRef]
7. Bashi-Azghadi, S.N.; Afshar, M.H.; Afshar, A. Multi-objective optimization response modeling to contaminated water distribution networks: Pressure driven versus demand driven analysis. *KSCE J. Civ. Eng.* **2017**, *21*, 2085–2096. [CrossRef]
8. Sousa, J.; Muranho, J.; Sá Marques, A.; Gomes, R. Optimal management of water distribution networks with simulated annealing: The c-town problem. *J. Water Resour. Plan. Manag.* **2016**, *142*, C4015010. [CrossRef]
9. Mashford, J.; De Silva, D.; Marney, D.; Burn, S. An approach to leak detection in pipe networks using analysis of monitored pressure values by support vector machine. In Proceedings of the 2009 Third International Conference on Network and System Security, Gold Coast, QLD, Australia, 19–21 October 2009 ; IEEE: Piscataway, NJ, USA, 2009; pp. 534–539.
10. Wu, Y.; Liu, S. A review of data-driven approaches for burst detection in water distribution systems. *Urban Water J.* **2017**, *14*, 972–983. [CrossRef]
11. Moser, G.; Paal, S.G.; Smith, I.F. Leak detection of water supply networks using error-domain model falsification. *J. Comput. Civ. Eng.* **2018**, *32*, 04017077. [CrossRef]
12. Van der Walt, J.; Heyns, P.S.; Wilke, D.N. Pipe network leak detection: Comparison between statistical and machine learning techniques. *Urban Water J.* **2018**, *15*, 953–960. [CrossRef]
13. Zhou, X.; Tang, Z.; Xu, W.; Meng, F.; Chu, X.; Xin, K.; Fu, G. Deep learning identifies accurate burst locations in water distribution networks. *Water Res.* **2019**, *166*, 115058. [CrossRef] [PubMed]
14. Sophocleous, S.; Savić, D.; Kapelan, Z. Leak localization in a real water distribution network based on search-space reduction. *J. Water Resour. Plan. Manag.* **2019**, *145*, 04019024. [CrossRef]
15. Huang, Y.; Zheng, F.; Kapelan, Z.; Savic, D.; Duan, H.F.; Zhang, Q. Efficient leak localization in water distribution systems using multistage optimal valve operations and smart demand metering. *Water Resour. Res.* **2020**, *56*, e2020WR028285. [CrossRef]
16. Fan, X.; Zhang, X.; Yu, X.B. Machine learning model and strategy for fast and accurate detection of leaks in water supply network. *J. Infrastruct. Preserv. Resil.* **2021**, *2*, 1–21. [CrossRef]
17. Guo, Z.; Leitao, J.P.; Simões, N.E.; Moosavi, V. Data-driven flood emulation: Speeding up urban flood predictions by deep convolutional neural networks. *J. Flood Risk Manag.* **2021**, *14*, e12684. [CrossRef]
18. Li, Z.; Wang, J.; Yan, H.; Li, S.; Tao, T.; Xin, K. Fast Detection and Localization of Multiple Leaks in Water Distribution Network Jointly Driven by Simulation and Machine Learning. *J. Water Resour. Plan. Manag.* **2022**, *148*, 05022005. [CrossRef]
19. Huang, L.; Du, K.; Guan, M.; Huang, W.; Song, Z.; Wang, Q. Combined Usage of Hydraulic Model Calibration Residuals and Improved Vector Angle Method for Burst Detection and Localization in Water Distribution Systems. *J. Water Resour. Plan. Manag.* **2022**, *148*, 04022034. [CrossRef]
20. Daniel, I.; Pesantez, J.; Letzgus, S.; Khaksar Fasaee, M.A.; Alghamdi, F.; Berglund, E.; Mahinthakumar, G.; Cominola, A. A Sequential Pressure-Based Algorithm for Data-Driven Leakage Identification and Model-Based Localization in Water Distribution Networks. *J. Water Resour. Plan. Manag.* **2022**, *148*, 04022025. [CrossRef]
21. Hu, Z.; Chen, B.; Chen, W.; Tan, D.; Shen, D. Review of model-based and data-driven approaches for leak detection and location in water distribution systems. *Water Supply* **2021**, *21*, 3282–3306. [CrossRef]
22. Simpson, A.; Elhay, S. Jacobian matrix for solving water distribution system equations with the Darcy-Weisbach head-loss model. *J. Hydraul. Eng.* **2011**, *137*, 696–700. [CrossRef]
23. Rossman, L.A. EPANET 2. In *Users Manual*; US Environmental Protection Agency (EPA): Washington, DC, USA, 2000.
24. Crowl, D.A.; Louvar, J.F. *Chemical Process Safety: Fundamentals with Applications*; Pearson Education: London, UK, 2001.
25. Klise, K.; Hart, D.; Bynum, M.; Hogge, J.; Haxton, T.; Murray, R.; Burkhardt, J. *Water Network Tool for Resilience (WNTR) User Manual*; Technical Report; Sandia National Lab. (SNL-NM): Albuquerque, NM, USA, 2020.

26. Montgomery, D.C.; Peck, E.A.; Vining, G.G. *Introduction to Linear Regression Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2021.
27. Hagberg, A.; Conway, D. Networkx: Network Analysis with Python. 2020. Available online: https://networkx.github.io (accessed on 8 January 2010).