Supervised Machine Learning Approaches for Leak Localization in Water Distribution Systems: Impact of Complexities of Leak Characteristics

Lochan Basnet[1]; Downey Brill, M.ASCE[2]; Ranji Ranjithan[3]; and Kumar Mahinthakumar, M.ASCE[4]

Abstract: Localizing pipe leaks is a significant challenge for water utilities worldwide. Pipe leaks in water distribution systems (WDSs) can cause the loss of a large amount of treated water, leading to pressure loss, increased energy costs, and contamination risks. What makes localizing pipe leaks challenging is the underground location of the water pipes and the similarity in impact on hydraulic properties (e.g., pressure, flow) due to leaks as compared to the effects of WDS operational changes. Physical methods to locate leaks are expensive, intrusive, and heavily localized. Computational approaches such as data-driven machine learning models provide an economical alternative to physical methods. Machine learning models are readily available and easily customizable to most problems; therefore, there is an increasing trend in their application for leak localization in WDSs. While several studies have applied machine learning models to localize leaks in single pipes and small test networks, these studies have yet to thoroughly test these models against the different complexities of leak localization problems, and hence their applicability to real-world WDSs is still unclear. The simplicity of the WDSs, the oversimplification of leak characteristics, and the lack of consideration of modeling and measuring device uncertainties adopted in most of these studies make the scalability of their proposed approaches questionable to real-world WDSs. Our study addresses this issue by devising four study cases of different complexity that account for realistic leak characteristics and model- and measuring device-related uncertainties. Two established machine learning models—multilayer perceptron (MLP) and convolutional neural network (CNN)—are trained and tested for their ability to localize the leaks and predict their sizes for each of the four study cases using different simulated hydraulic inputs. In addition, the potential benefit of combining different types of hydraulic data as inputs to the machine learning models in localizing leaks is also explored. Pressure and flow, two common hydraulic measurements, are used as inputs to the machine learning models. Further, the impact of single and multiple time point input in leak localization is also investigated. The results for the L-Town network indicate good accuracies for both the models for all study cases, with CNN consistently outperforming MLP. **DOI:** **10.1061/JWRMD5.WRENG-6047.** *© 2023 American Society of Civil Engineers.*

Introduction

Localizing pipe leaks is a significant challenge for water utilities worldwide. An estimated 126 billion cubic meters of water is lost worldwide through water distribution systems (WDSs) due to pipe leaks and breaks yearly (Liemberger and Wyatt 2019), which causes billions of dollars in revenue loss. Further, it jeopardizes the growing imbalance between water supply and demand, especially in the face of water scarcity and climate change. Pipe leaks can also pose risks of water contamination (Fontanazza et al. 2015), property damage, and traffic disruptions. Therefore, timely localization and prevention of pipe leaks are paramount. However, pipe leaks are challenging to locate because their impact on hydraulic properties such as flow and pressure are not easily discernible (Yu et al. 2021) and can be similar to the effects of WDS operational changes. In addition, the pipes being underground makes it even harder to locate leaks. Field methods to locate leaks are expensive and can cause interruption to water service (Rajeswaran et al. 2018). Computational approaches provide an economical alternative to field methods [an extensive review of the computational approaches for leak characterization is provided by Hu et al. (2021)]. Machine learning approaches are one of the data-driven computational approaches that have gathered increasing interest in the last two decades in leak characterization studies (Zaman et al. 2020). Data related to the hydraulic properties of WDSs, such as pressure, flow, acoustics, optics, or temperature, can be used to localize leaks. Pressure and flow are the most common measurements used with machine learning approaches for leak localization (Abdulla and Herzallah 2015).

While many research studies on applying machine learning approaches for leak characterization (i.e., detection and localization) in pipes have been conducted (Wu and Liu 2017), their applicability and effectiveness toward the different complexities of leak characterization problems are yet to be fully tested. This lack of

[1] Ph.D. Student, Dept. of Civil Engineering, North Carolina State Univ., Raleigh, NC 27606 (corresponding author). ORCID: https://orcid.org/0000-0002-4698-4426. Email: lbasnet@ncsu.edu

[2] Professor, Dept. of Civil Engineering, North Carolina State Univ., Raleigh, NC 27606. Email: brill@ncsu.edu

[3] Professor, Dept. of Civil Engineering, North Carolina State Univ., Raleigh, NC 27606. Email: ranji@ncsu.edu

[4] Professor, Dept. of Civil Engineering, North Carolina State Univ., Raleigh, NC 27606. ORCID: https://orcid.org/0000-0002-9852-1888. Email: gmkumar@ncsu.edu

Note. This manuscript was submitted on November 18, 2022; approved on March 23, 2023; published online on May 24, 2023. Discussion period open until October 24, 2023; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Water Resources Planning and Management***, , ISSN 0733-9496.

thorough testing makes the applicability of these approaches to real-world WDSs unclear. One of the complexities of leak characterization problems involves the size of the WDS, which dictates the scalability of the applied models to real-world WDSs. Zhou et al. (2019a) and Shukla and Piratla (2020) used convolutional neural network (CNN) models for leak detection in a single pipe using simulated negative pressure wave and scalogram images of vibration signals as inputs, respectively. Pérez-Pérez et al. (2021) used a multilayer perceptron (MLP) model with a cascade-forward back-propagation to detect leaks in a single pipe using simulated pressure data. However, analyzing leaks by isolating individual pipes in complex interconnected WDSs is not viable in the field because it is difficult to isolate specific pipes. Further, the tools and resources required to collect some of these input data for individual pipes in large real-world WDSs are infeasible. Beyond single-pipe analyses, several studies have considered complete or partial hydraulic systems. Bohorquez et al. (2020) used an MLP to predict leaks in a simple hydraulic system using numerically obtained fluid transient waves as input. Mashford et al. (2012) used a support vector machine to predict leak size and location for an isolated section of a WDS based on simulated pressure data. Soldevila et al. (2016) used a model-based *k*-nearest neighbors (k-NN) classifier to identify leak events and locations. Lučin et al. (2021) used a random forest model to locate leaks in two small-sized networks. These studies still face the challenge of scalability because extrapolating their results and, therefore, application to larger WDS networks typical of real-world systems is very challenging.

Another key complexity of the leak localization problem is associated with the characteristics of pipe leaks. In real-world WDSs, multiple leaks of varying sizes can occur simultaneously at different locations within the WDS, making leak localization very challenging. Furthermore, the uncertainties associated with hydraulic simulation models and the imprecision of measurement devices in real-world WDSs add more complexity to the leak localization problem. For example, the parameters such as demands, pipe roughness, pipe diameters, and lengths used in the hydraulic models have associated uncertainties (Blesa and Pérez 2018). These uncertainties affect the accuracy of the simulated pressure and flow data. Zhou et al. (2019b) account for the hydraulic model parameter uncertainties by adding noise to these parameters prior to simulation. However, their approach does not account for the uncertainties related to the imprecision of measurement devices such as pressure sensors and flow meters of the real-world WDSs. If some knowledge about the anticipated uncertainties is available, we can train machine learning approaches using simulated data corrupted with noise that takes into account these uncertainties, as shown by Soldevila et al. (2017), Mohammed et al. (2021), and Santos-Ruiz et al. (2020). While these studies have successfully identified the different forms of the leak localization problem complexities, they failed to analyze them comprehensively. These complexities are not mutually exclusive and are highly likely to manifest together at once. Therefore, any selective analyses that assume the presence of one complexity and the absence of another is inadequate. Soldevila et al. (2017) and Santos-Ruiz et al. (2020) are limited to the analysis of single leaks only; they fail to provide any understanding of how their approach would perform for multiple leaks. Mohammed et al. (2021) considers single leak only for cases with uncertainties and implicitly assumes the absence of any form of uncertainties for multiple leak scenarios. Furthermore, all of these studies use a small-sized network in their analysis, therefore, suffer from scalability, as aforementioned.

Leak characterization studies that rely on a hydraulic model use conventional optimization approaches (i.e., genetic algorithms, quadratic programming, linear programming, etc.) that seek to minimize the differences between measured and simulated pressures or flows (e.g., Berglund et al. 2017; Ponce et al. 2014; Steffelbauer and Fuchs-Hanusch 2016; Wu et al. 2010). However, this is an inverse problem and can suffer from ill-posedness under noisy conditions if the leak signature is weak or non-existent at one or more sensors. In contrast, machine learning models are trained to minimize the difference between actual and predicted leak values and can learn to disregard false pressure and flow responses that do not correlate with leaks. This ability to learn under noisy conditions is a significant advantage of machine learning methods over conventional approaches. When it comes to the choice of machine learning models, depending on their architecture, way of handling different inputs, and ability to learn spatial and temporal patterns, different machine learning models may perform differently for the same level of complexities of leak localization problems. A comparison of the performance of machine learning models that vary in architecture, input handling, and learning ability for the different problem complexities can provide clarity.

Different hydraulic measurements can provide different information regarding leaks; therefore, there is a possibility to improve leak localization performance if multiple hydraulic data are used. However, to the authors' knowledge, studies concerning machine learning approaches have yet to explore this possibility. A quick way to test this possibility is by analyzing the leak localization performance of machine learning models trained with a single input to the models trained with two simultaneous inputs. For example, pressure and flow can be used as two inputs because they have been used widely in leak localization studies. Mohammed et al. (2021) used pressure and flow inputs in their statistical approach for leak localization. However, their study did not analyze the impact of additional flow data on the accuracies obtained using pressured data only. Finally, for time series inputs such as pressure and flow, the number of time points can also impact the performance of machine learning models in localizing leaks. Therefore, investigation of leak localization performances for single and multiple time point data is also necessary.

This study proposes a machine learning–based approach to evaluate the impact of the different complexities of leak localization problems on the applicability and effectiveness of the prediction models. This approach compares two different readily available and widely used neural network–based models, examines the benefit of combining different types of hydraulic data (i.e., pressure and flow), considers multiple realistic leak scenarios, and accounts for hydraulic model uncertainties and instrument imprecision. The key contributions of this study with respect to previous studies include the following:

1. Comprehensively analyzing the various complexities of leak localization problem by:
   - Overcoming the unrealistic simplification; that is, the occurrence of a single leak at a time assumed in most state-of-the-art techniques (Soldevila et al. 2017) by generalizing it to multileak problems.
   - Considering leaks of varying sizes to represent more realistic leak scenarios.
   - Accounting for the realistic nature of leak locations by considering the possibility of random leak locations anywhere within a WDS.
   - Consideration of the most common and impactful hydraulic model uncertainty; that is, demand uncertainty, as well as measuring instrument imprecision through the addition of noise to the input data.
2. Analysis of the interplay of the complexities of the leak localization problem with the different machine learning models.

3. Research on the implication of simultaneous use of multiple hydraulic data and the impact of additional time points on leak localization.
4. Simultaneous prediction of location as well as the size of the leaks. Leak size information can be beneficial for decision-making on repairs and replacement, especially when considering budgeting and resource constraints.

Even though the data used to train the machine learning models are simulated data, they are applicable to localize leaks using real-world measurements as long as the hydraulic model reasonably represents the actual WDS. However, if the existing hydraulic model is calibrated with background leaks, any approach based on the simulated hydraulic data can only be used to locate new leaks. On the other hand, for WDSs that have abundant real-world pressure and flow sensor measurements, these models can easily be fine-tuned and tested using real data.

Methods

General Framework

Fig. 1 illustrates the general framework proposed in this study to localize leaks in WDS pipes. The framework starts with a WDS hydraulic model that generates simulated pressure and flow data. First, pressure and flow data are generated for a leak-free scenario by simulating the hydraulic model using the EPANET simulator (Eliades et al. 2016). It is then followed by generating pressure and flow data for multiple different leak scenarios. Pressure and flow differences between the leak and leak-free scenarios are then computed and stored as pressure and flow data, respectively. The corresponding leak scenarios are stored as a leak values data set. Next, noise is added to the pressure and flow data sets as required by the case under study described in the "Study Cases" section. The data sets are then randomly shuffled and split into training and testing sets. The training data sets consist of the pressure and flow data (covariates) and the leak values (response). These data are scaled and fed to the machine learning models for training and tuning, and the optimized models are selected for localizing the leaks. As a final step, the leak prediction (location and size of the leaks) and model evaluation are performed on the test data sets using the optimized models; predicted model outputs are compared with the corresponding true leak values.

Machine Learning Models

Multilayer Perceptron

MLP models are supervised-learning models based on deep neural networks. MLP models consist of an input layer, an output layer, and a selected number of dense hidden layers between the input and the output layers (Fig. 2). Each unit of a hidden layer consists of an activation subunit that activates or deactivates the received signals (Fig. 3). Multiple activation functions are available to be used within these activation subunits.

One-Dimensional Convolutional Neural Network

Like MLP models, CNN models are supervised learning–based deep neural networks. The key difference between CNN and MLP models is the presence of convolutional and pooling layers in CNN models. As shown in Fig. 4, the convolutional layers produce convolved feature maps, which allow for contextual learning, and the pooling layers downsample these maps to extract abstract

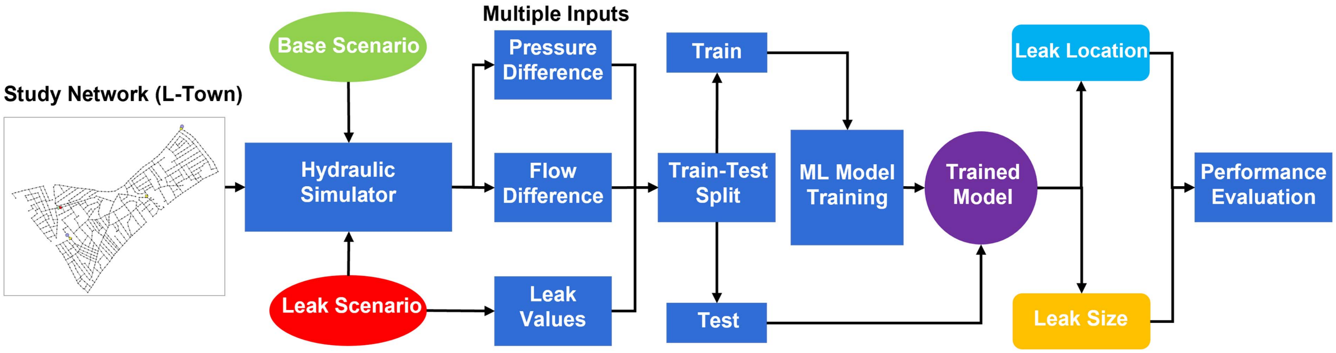

Fig. 1. General framework for detecting leaks in WDS.

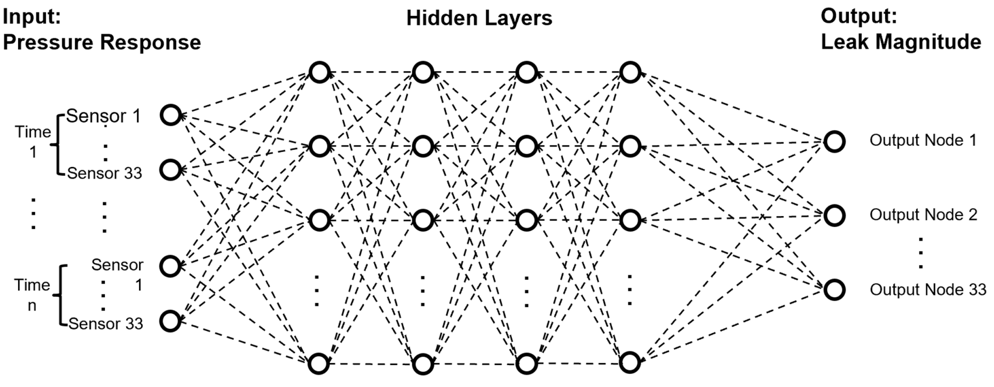

Fig. 2. Multilayer perceptron.

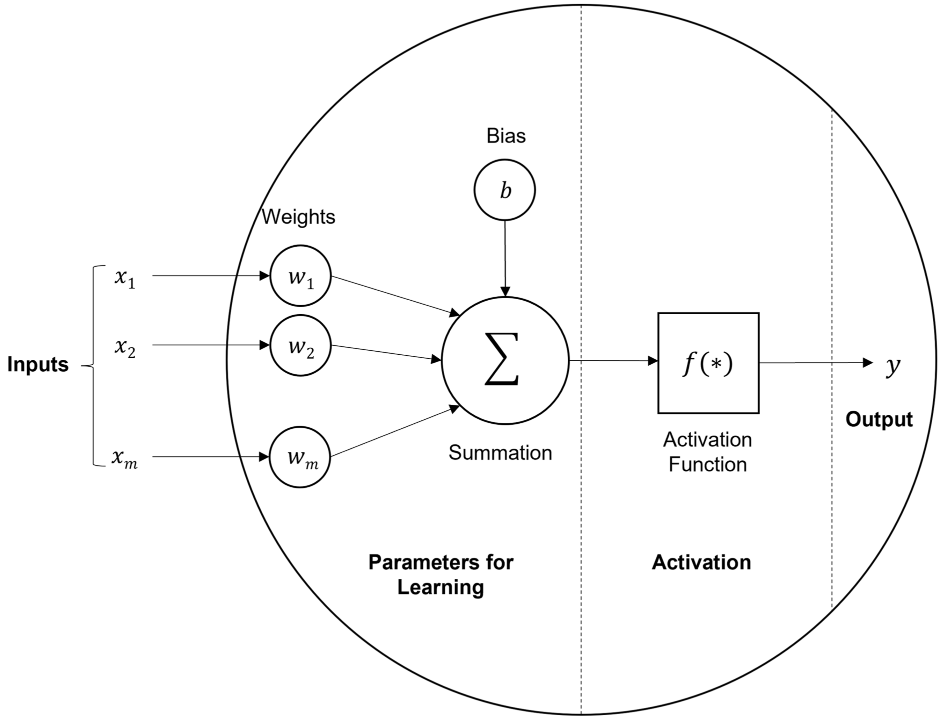

Fig. 3. Individual unit of a hidden layer.

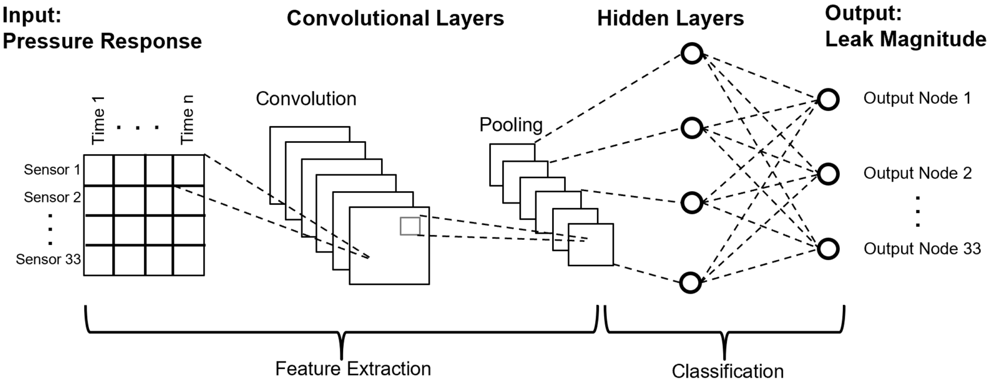

Fig. 4. Convolutional neural network.

features from the data. The convolutional layers use kernels or filters to extract the features. A one-dimensional (1D) CNN model uses filters that only vary in depth (i.e., one dimension). Like MLP models, 1D CNN models also consist of an input layer, an output layer, dense hidden layers, and activation layers.

Multiple Input Models

Deep neural network–type machine learning models are generally built using a sequential architecture where one hidden layer is built on top of another (i.e., linear). Sequential-type architectures limit the data flow into the models and restrict these models to only a single input. However, some machine learning model packages provide additional nonlinear-type architectures. For example, the Keras package of TensorFlow (Abadi et al. 2016) in Python used in this study provides a different functional-type architecture where multiple sets of hidden layers can be built independently and combined later, as required. This flexibility of functional-type architecture means multiple types of inputs can be used simultaneously to train machine learning models. Fig. 5 shows the architecture of the CNN and MLP models developed for this study; the models use pressure and flow as two simultaneous inputs. The two inputs train the models independently before being combined into a single training construct. This model construct is particularly advantageous for fully connected models such as the MLP, where one perceptron can affect the learning of all other perceptrons and in cases where one input may be uninformative or noisy compared to the other. In the latter case, as the uninformative/noisy data goes through a set of layers, the uninformative data fails to activate the perceptrons and dies out; thus, preventing any carryover that might adulterate the other informative input data.

Hyperparameters and Model Tuning

Total Number of Iterations (Epochs)

MLP and CNN models are trained for several iterations (epochs) to ensure stability in the training process. The optimal model and its

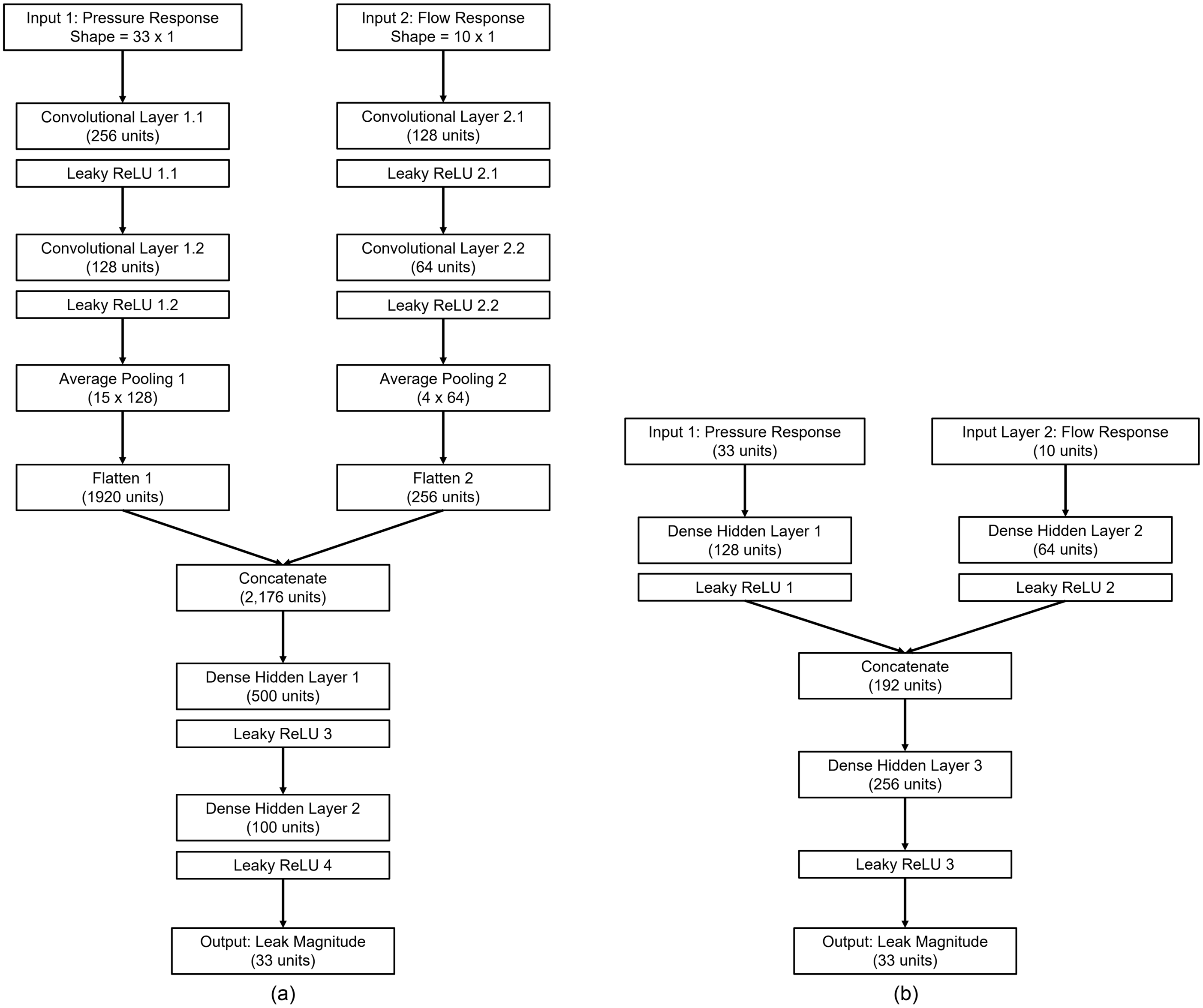

Fig. 5. Modified architecture for input data consisting of multiple types: (a) CNN; and (b) MLP.

corresponding weights are determined by monitoring the training and validation errors over the entire number of epochs.

Loss Function

The functions to calculate the training and validation errors are chosen based on the nature of the problem. In this study, leak localization is formulated as a regression-type problem to solve simultaneously for both leak locations and sizes. Therefore, the mean square error (MSE) function is used; mean absolute error (MAE) can be used as an alternative to MSE. The MSE loss function minimized during training to determine the optimal model is

$$\mathrm{MSE} = \sum_{j=1}^{S} \sum_{i=1}^{33} (X_{ij}^{P} - X_{ij}^{A})^2 \tag{1}$$

where X_{ij}^{P} is the model predicted leak size for ith candidate leak region of jth leak scenario, and X_{ij}^{A} is the actual leak size for ith candidate leak region of jth leak scenario.

Activation Function

A trial-and-error evaluation of multiple activation functions identified the leaky rectified linear unit (L-ReLU) (Maas and Ng 2013) as a suitable activation function for this study. L-ReLU prevents the problem of vanishing gradients during forward propagation like the regular rectified linear unit (ReLU) and has the added advantage of preventing vanishing gradients during backward propagation (Goodfellow et al. 2016).

Optimizer

The Adam algorithm (Kingma and Ba 2014) is used as the optimizer in this study.

Study Network

In this study, the leak localization models are applied to a standard test network called the L-Town water network (Fig. 6). The L-Town network is a city-scale model based on a coastal city in Cyprus. This network has been previously used in several modeling and simulation-related researches. For example, this network was also

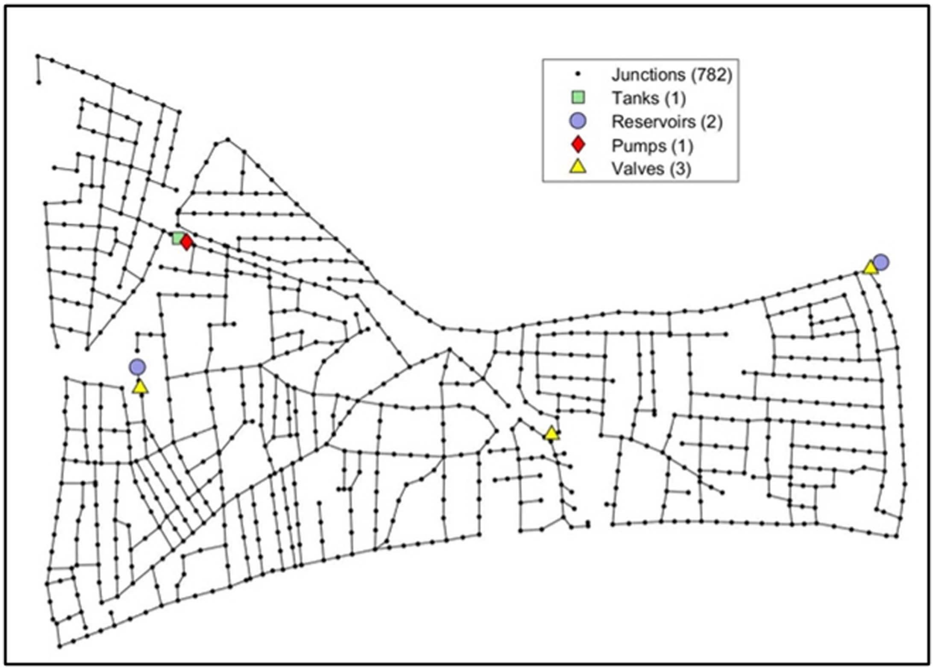

Fig. 6. L-Town water network.

used in the leakage detection and isolation methods (BattLeDIM 2020; Vrachimis et al. 2020) competition to evaluate the performances of different machine learning and computational models for leak detection. The L-Town network consists of 905 pipes and 782 junctions and is primarily a tank-regulated model network.

Candidate Leak Regions

Localizing leaks to the pipe- or junction-level requires a large amount of data, which is infeasible to obtain from real-world WDSs. Therefore, a lesser resolution needs to be adopted for leak localization. This study divided the entire L-Town water network into several subareas considered candidate leak regions. The L-Town network is divided into 33 candidate leak regions [Fig. 7(a)] using a *k*-means clustering technique (Lloyd 1982) based on Euclidean distances. A hydraulic distance-based clustering, applied by Zhang et al. (2016), is not used in this study because it resulted in noncontiguous ununiformly sized clusters. For practitioners, such noncontiguous and ununiform clusters are impractical to be used in the field. Further, the clusters obtained using Euclidean distance-based measures are hydraulically less homogeneous and pose a more significant challenge for the leak localization models. Therefore, the Euclidean distance-based clustering used here is a more conservative approach.

In this study, leaks are modeled as emitters using EPANET and are assumed to occur at the center of the pipes. Because EPANET supports emitters only on nodes, new junction nodes are inserted in the middle of the leaky pipe in the network using the Morph

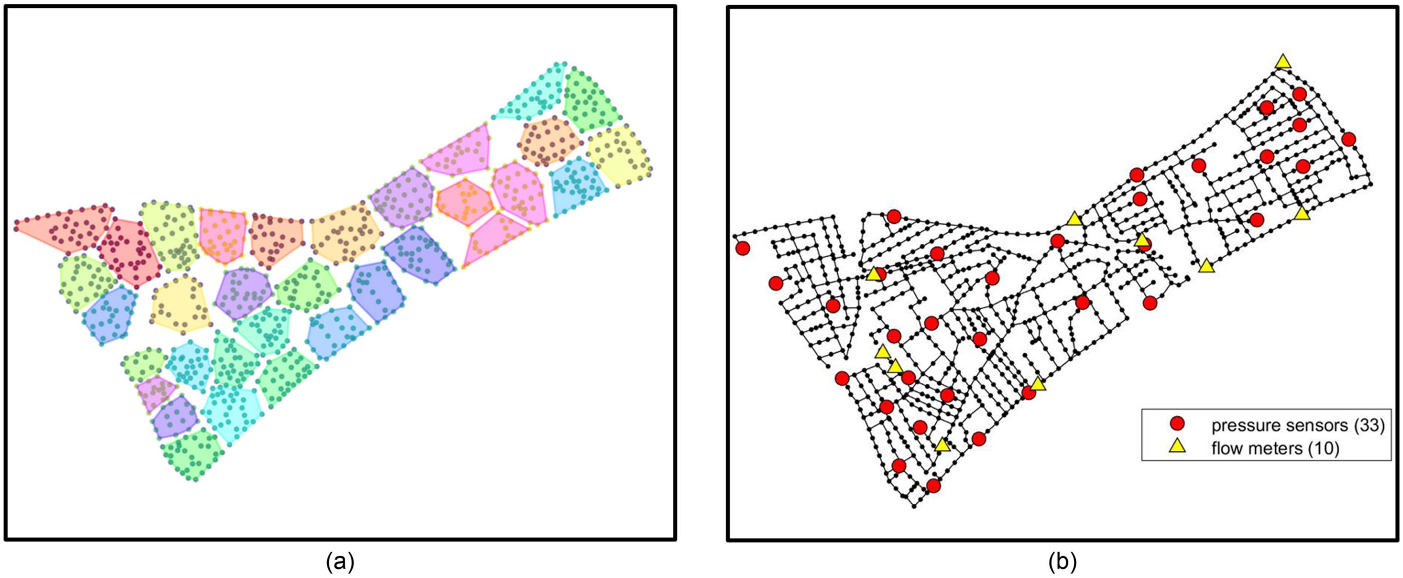

Fig. 7. Candidate leak regions: (a) 33 leak regions; and (b) pressure (circle) and flow (triangle) sensors.

package in WNTR (Klise et al. 2017). Candidate leak nodes representing each leak region are assumed to be at the centroid of each leak region. Centroids of leak regions are estimated using the k-NN search algorithm. For any given leak scenario, this region defines a leak located anywhere within the boundaries of a candidate leak region.

A pressure node is assigned to each of the 33 candidate regions to track the pressure changes due to a leak in that region. These pressure nodes represent pressure sensors in real-world WDSs. The locations of the pressure nodes are based on the locations used in BattLeDIM 2020 and are shown in Fig. 7(b). Only 10 flow meters are assumed to be located in the WDS. Fig. 7(b) shows the location of these flow meter nodes.

Study Cases

Four study cases are considered to represent the complexities associated with real leak characteristics and the uncertainties in input data due to water network model inaccuracies and measuring device imprecision.

Case A: No-Noise

Input pressure and flow difference data are free of noise. The *no-noise* case represents the ideal but unrealistic case with accurate WDS models and precise measuring devices. Leaks are assumed to occur at the centroid of each leak region.

Case B: Demand-Noise

Input data accounts for the WDS model inaccuracies. Random Gaussian noise is added to the demands prior to simulation to mimic the inaccuracies in demand values in the WDS model. While exponential-type noise can better capture the pulsed nature of demand, the Gaussian-type noise was found to affect the model performance more. Simulated pressure and flow data are then generated using the modified WDS model. A 10% Gaussian noise is added to the demand parameters. Leaks are assumed to occur at the centroid of each leak region.

Case C: Mixed-Noise

Input data accounts for the WDS model inaccuracies and the measuring device imprecision. Leaks are assumed to occur at the centroid of each leak region. Noise is added to the pressure and flow differences between the leak and leak-free scenarios after the hydraulic simulations. A 10% Gaussian noise is added to the pressure and flow differences. An analysis of the coverage of error residuals due to a 10% Gaussian noise indicated that it accounts for error residuals due to a combination of demand parameter uncertainty (10%), pipe roughness coefficient uncertainty (±1), and node elevation uncertainty (±0.15 m) in addition to any anticipated measurement errors. Moser et al. (2018) analyzed these three model parameter uncertainties individually.

Case D: Random Leaks

Unlike the previous three cases, the leaks are not fixed at the centroid of the leak regions. Instead, the leaks can be located anywhere within the boundaries of the candidate leak regions. Two or more leaks located anywhere within the same leak region are labeled identically. No additional noise is imposed on the hydraulic inputs (flow and pressure).

Data Generation

The input data sets used in this study constitute the leak scenario and the pressure difference data sets, which are generated sequentially in the following order.

Leak Scenario Generation

The following four assumptions are considered for the generation of realistic leak scenarios in this study:

- A leak scenario must consist of at least 1 leak.
- A leak scenario can include a maximum of 3 leaks.
- A leak can be located in any of the 33 candidate leak regions.
- The leak size ranges from 0 to 5 as compared to the 0 to 3 range used in BattLeDIM 2020.

The leak size is the discharge/emitter coefficient in the leak

$$q = Cp^{\gamma} \tag{2}$$

where q is the flow rate, p is the pressure, C is the discharge/emitter coefficient, and Υ (= 0.5) is the pressure exponent.

Applying these assumptions, leak scenarios are generated using the following general procedure:

Step 1. For a leak scenario, the total number of leaks is determined by randomly drawing a number n from the set {1, 2, 3}.

Step 2. Based on the outcome n of the previous draw, n candidate leak regions are drawn randomly out of the 33 candidate leak regions.

Step 3. For each of the n candidate leak regions, the leak size is randomly drawn from the 0 to 5 range.

Step 4. Store these leak sizes in a vector (size $= 33 \times 1$) at locations corresponding to their candidate leak region number. For example, the leak size for candidate region 1 is stored at vector position (1, 1). Candidate regions with no leak are assigned a value equal to zero.

Step 5. Repeat steps 1–4 for 100,000 times to generate 100,000 leak scenarios.

Step 6. Store the resulting 100,000 leak scenarios as a leak scenario data set (a matrix of size $= 100{,}000 \times 33$).

Pressure and Flow Data Generation

Simulated pressure data are generated by the following procedure:

Step 1. Simulate a leak-free scenario for the specified study case by running the base model with the EPANET simulator. Then, store the resulting pressure and flow at the 33 pressure nodes and the 10 flow sensor nodes, respectively.

Step 2. Pick a leak scenario from the leak scenario data set and add the associated leaks to the base model. Then, run this modified model with the EPANET simulator and store the resulting pressure and flow at the 33 pressure nodes and the 10 flow sensor nodes, respectively.

Step 3. Repeat Step 3 for all the 100,000 leak scenarios in the leak scenario data set.

Step 4. Compute the pressure and flow differences between the leak scenarios (100,000 scenarios) and the leak-free scenario to generate the pressure difference data set (matrix of size $=$ $100{,}000 \times 33$) and the flow difference data set (matrix of size $=$ $100{,}000 \times 10$), respectively.

Step 5. Add noise to the pressure and flow data sets if needed for the study case.

Model Validation and Testing

Train–Test Split

The pressure, flow, and leak data sets are divided into training and test data. A training–test ratio of 80:20 is used to split the data. The two models are validated using the test data sets.

Metrics and Thresholds

The performance of the two machine learning models in localizing the leaks is evaluated using the two standard classification metrics:

Table 1. Machine learning model details

Model	Architecture	Hidden layers	Convolution layers	Activation functions	Learning rate	Optimizer
CNN—single input	33-256-128-500-100-33	2	2	L-ReLU	0.05	Adam
CNN—two inputs	Fig. 4(a)	2	4	L-ReLU	0.05	Adam
MLP—single input	33-64-128-33	2	—	L-ReLU	0.05	Adam
MLP—two inputs	Fig. 4(b)	3	—	L-ReLU	0.05	Adam

precision and recall. In addition, their harmonic mean referred to as the F1-score is also calculated

$$\text{Precision}(P) = \frac{TP}{TP + FP} \times 100 \tag{3}$$

$$\text{Recall}(R) = \frac{TP}{TP + FN} \times 100 \tag{4}$$

$$F1 - \text{score}(F1) = \frac{2 \times P \times R}{P + R} \tag{5}$$

where TP is the true positives; FP is the false positives; and FN is the false negatives.

In the context of this study, precision is the percentage of actual leaks predicted correctly out of all leak predictions of the models. Recall is the percentage of actual leaks out of all the leaks in the data set predicted correctly by our models.

In this study, the problem of leak localization is formulated as a regression-type problem to solve simultaneously for both leak locations and sizes. Postprocessing of model outputs is required to calculate precision and recall. This postprocessing involves using thresholds to determine correct/incorrect location and size classifications. A set of nine thresholds ranging from 0.1 to 0.9, increasing incrementally by 0.1, are used. The thresholds are in the same unit as leak sizes and represent the precision of the measuring devices for real-world systems. For example, a threshold of 0.1 means that the leaks smaller than 0.1 in the data set are considered no-leaks, and only the predictions within 0.1 units of the actual leak values are considered correct classifications. A model prediction is assumed to be correct in both location and size if the position of the predicted leak size (within the 33×1 vector) is correct and the absolute difference between the predicted and the actual value is within the threshold.

Software and Tools

The following software and tools were used in this study:

- EPANET Simulator 2.0 version—Hydraulic simulations are performed using EPANET simulator.
- WNTR Morph package—For splitting the network to add junction nodes at the middle of each pipe.
- Matlab 2019b version—Input data generation is done by running the EPANET simulator in Matlab. Matlab is also used to generate candidate leak regions and nodes.
- Python version 3.7—Model training, testing, and validation is done in Python.
- Tensorflow version 2.1.6—Machine learning models are built using the Tensorflow package.

Results and Discussions

The leak prediction performances of the MLP and CNN models are evaluated for the four study cases. The two models are compared by calculating precision and recall accuracies for the test data set. In addition, with both models, the potential benefit of using multiple hydraulic properties as simultaneous inputs is investigated by comparing the model accuracies for a single input (pressure data) to two simultaneous inputs (pressure and flow data). Table 1 summarizes the architecture and hyperparameters for the optimal MLP and CNN models for the single and multiple input cases. Fig. 5 shows the architecture of the optimal multiple-input CNN and MLP models. All models use L-ReLU as the activation function and the Adam algorithm as the optimizer. This is because our preliminary experimentation indicated that the combination of L-ReLU and Adam optimizer performed better than other activation functions (e.g., sigmoid and ReLU) and optimizers (e.g., RMSprop, SGD) and their various combinations. Fig. 8 shows the trend of the training and validation MSEs for the *no-noise* case for the two models. The validation errors show a general decreasing trend that stops after the 100th epoch, indicative of model overfitting beyond 100 epochs. The same is true for the validation errors for the other three study cases. Therefore, the required number of iterations for all model training was set to 100 epochs.

Problem Complexity due to Real-World Leak Characteristics

The precision and recall for the four study cases (with both CNN and MLP) were compared to understand the complexity of the leak localization task due to real-world leak characteristics. The complete model performances for the MLP and CNN models for the four study cases (*no-noise*, *demand-noise*, *mixed-noise*, and *random leaks*) are summarized in Tables 2 and 3. The *no-noise* study case represents the ideal but unrealistic condition where the input data is noiseless and perfect. Both precision and recall at all thresholds for the *no-noise* case rank highest compared to the other three study cases for both MLP and CNN models. Precision and recall are comparatively high (>40%) even at the most stringent threshold (0.1) for the *no-noise* case. These high accuracies can be attributed to the fact that the leak signatures in the input pressure difference data that are key to locating leaks are unaffected in the absence of

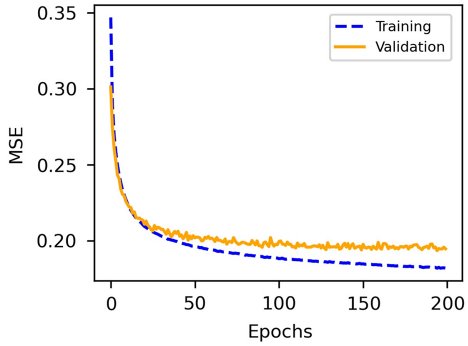

Fig. 8. Training and validation error dynamics.

Table 2. MLP model performance for the four study cases

	No-noise			Demand-noise			Mixed-noise			Random leaks		
Threshold	P	R	F1	P	R	F1	P	R	F1	P	R	F1
0.1	50.8	43.6	46.9	31.9	35.1	33.4	16.3	23.6	19.3	3.5	8.2	4.9
0.2	82.2	70.1	75.7	69.5	59.8	64.3	44.1	42.0	43.0	11.7	16.1	13.6
0.3	91.1	82.8	86.8	83.8	73.8	78.5	62.9	55.2	58.8	21.3	23.7	22.4
0.4	94.8	89.6	92.1	89.9	82.4	86.0	73.4	64.4	68.6	31.2	31.1	31.1
0.5	96.5	93.2	94.8	93.0	87.8	90.3	79.9	71.2	75.3	40.0	37.7	38.8
0.6	97.4	95.4	96.4	94.9	91.2	93.0	84.1	76.4	80.1	48.1	44.1	46.0
0.7	98.0	96.8	97.4	96.0	93.4	94.7	87.1	80.4	83.6	54.9	49.6	52.1
0.8	98.4	97.5	97.9	97.0	95.0	96.0	89.3	83.4	86.2	60.7	54.5	57.4
0.9	98.7	98.0	98.3	97.6	96.2	96.9	91.0	85.9	88.4	65.8	58.9	62.2

Note: P = precision; R = recall; and F1 = F1-score.

noise. The *demand-noise* case considers the possibility of uncertainty in the demand parameters of hydraulic models. It ranks second among the four study cases based on precision and recall values. While the uncertainty in demand parameters in the hydraulic model can generate noise in the simulated pressure data, the noise is not comparatively large and minimally affects the leak signatures. For the *mixed-noise* case, a 10% Gaussian noise is added to the input pressure differences to account for measurement imprecision and model uncertainties. The noise introduces randomness in the data, affecting the leak signatures to a higher degree. Therefore, the precision and recall of the two models for the *mixed-noise* case are considerably low compared to *no-noise* and *demand-noise* cases. The leak signatures are affected to the highest degree for the *random leaks* case; that is, the fourth study case. While no direct noise is added to the input data, the randomness in leak locations within a candidate leak region introduces the possibility of a multitude of unique leak signatures for multiple leak scenarios considered the same (i.e., labeled identically); as defined previously, leaks located differently within the same candidate leak region are labeled identically. The degree of dissimilarity of the different leak signatures for identically labeled leak scenarios is widened further because the candidate regions are Euclidean distance-based clusters and not hydraulically similar clusters. Therefore, the *random leaks* case presents the most challenging task for machine learning models to learn. Consequently, the precision and recall accuracies for the *random leaks* case are the lowest. The effect of the complexity of the *mixed-noise* and the *random leaks* cases is profound at the lower thresholds, especially at 0.1, because the noise and randomness in the input data drown out the leak signature in the pressure input resulting from such small leak sizes or leak size differences.

Comparison of CNN and MLP Model Performance

Tables 2 and 3 summarize the model performances of the MLP and the CNN model for the four study cases at all nine thresholds considered in this study. Figs. 9–12 compare precision and recall for the two models at three selected thresholds (0.1, 0.5, and 0.9) for the four study cases. The results at these three thresholds represent all nine thresholds, with 0.1, 0.5, and 0.9 indicating the most-, the mild-, and the least-stringent conditions, respectively.

The figures show that precision for the CNN model is higher than the MLP model at all thresholds for all four study cases. At the 0.1 thresholds, the precision for CNN is almost twice the precision for MLP for all study cases. For the *no-noise* and *demand-noise* cases, the precision for CNN is more than 30% higher than the precision for MLP. The difference, however, starts to diminish as the threshold becomes less stringent (0.5 and higher thresholds) and the leak localization problem becomes comparatively less complex. The higher precision for the CNN model compared to the MLP model for all four study cases indicates its superiority in minimizing false leak predictions and hence better reliability in its predictions even when noise and randomness are present in the input data.

Like precision, recall for the CNN model is higher than the MLP model at all thresholds for all four study cases. However, while recall for CNN is considerably higher than for MLP at the 0.1 thresholds, the difference is not as high as the difference in precision. At the less stringent thresholds, particularly at 0.9, the difference in recall for the two models is almost negligible for the simple *no-noise* and the less complex *demand-noise* cases. However, the difference is greater (5%–12%) for the challenging *mixed-noise* case and even more so for the most challenging *random leaks* case at all thresholds, with the CNN model outperforming the MLP

Table 3. CNN model performance for the four study cases

	No-noise			Demand-noise			Mixed-noise			Random leaks		
Threshold	P	R	F1	P	R	F1	P	R	F1	P	R	F1
0.1	82.1	65.5	72.9	69.6	55.0	61.4	28.6	28.1	28.3	6.8	13.4	9.0
0.2	94.2	86.1	90.0	89.9	78.3	83.7	61.6	49.1	54.6	24.0	25.1	24.5
0.3	96.6	92.7	94.6	94.4	87.7	90.9	75.2	62.9	68.5	40.4	35.5	37.8
0.4	97.8	95.5	96.6	96.4	92.3	94.3	82.2	71.6	76.5	52.2	44.3	47.9
0.5	98.3	97.0	97.6	97.4	94.8	96.1	86.2	77.7	81.7	60.7	51.4	55.7
0.6	98.7	97.8	98.2	98.1	96.2	97.1	88.9	81.9	85.3	66.9	57.5	61.8
0.7	98.9	98.3	98.6	98.5	97.2	97.8	90.8	85.2	87.9	71.7	62.6	66.8
0.8	99.1	98.7	98.9	98.7	97.8	98.2	92.3	87.6	89.9	75.4	66.8	70.8
0.9	99.2	98.9	99.0	99.0	98.3	98.6	93.4	89.5	91.4	78.5	70.5	74.3

Note: P = precision; R = recall; and F1 = F1-score.

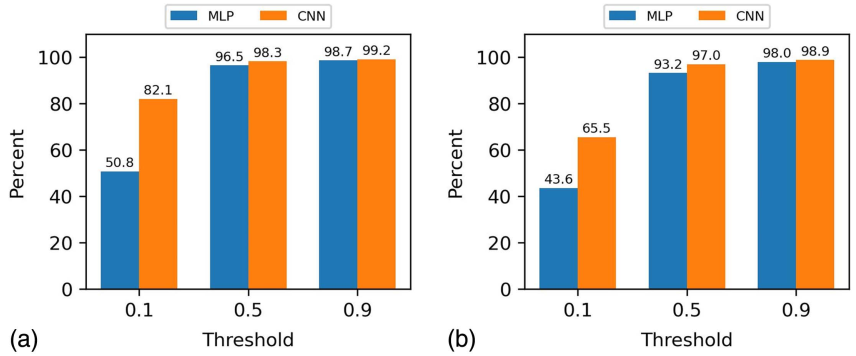

Fig. 9. Comparative model performance (CNN versus MLP) for no-noise case: (a) precision; and (b) recall.

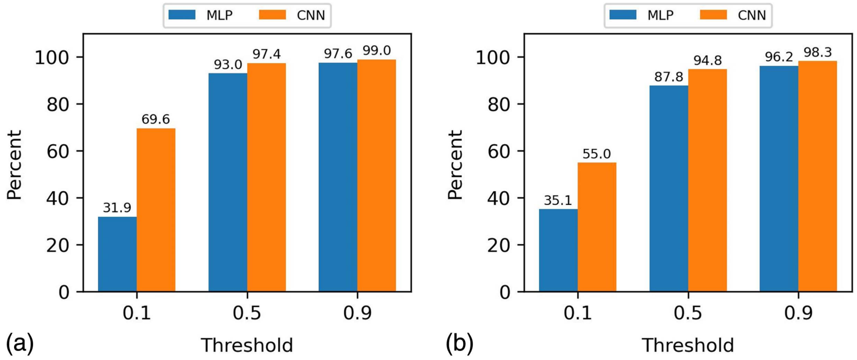

Fig. 10. Comparative model performance (CNN versus MLP) for demand-noise case: (a) precision; and (b) recall.

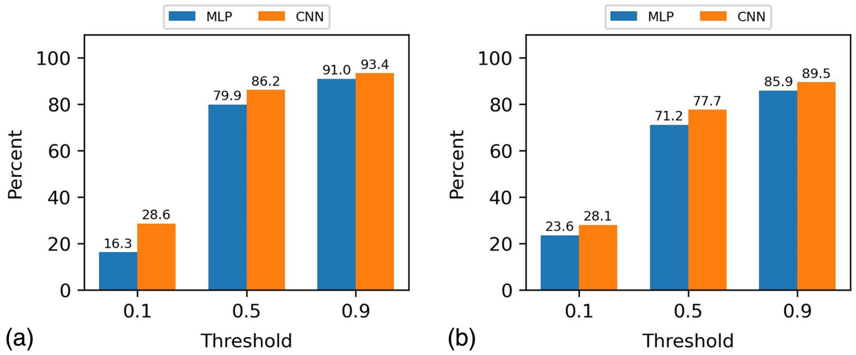

Fig. 11. Comparative model performance (CNN versus MLP) for mixed-noise case: (a) precision; and (b) recall.

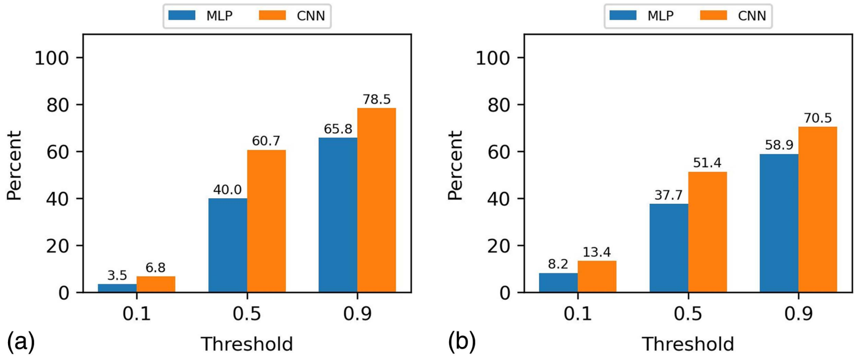

Fig. 12. Comparative model performance (CNN versus MLP) for random leaks case: (a) precision; and (b) recall.

model. Overall, the recall results are consistent with the precision results in implying the superior performance of the CNN model over the MLP for leak localization for the L-Town network.

The superiority of the CNN model to the MLP model can be attributed to several reasons. First, CNN models can account for any spatial relationships present in the data (LeCun et al. 1998). MLP models cannot do so because the inputs to MLP models are required to be flattened into vectors. Each vector element is then fed to a different perceptron (single unit). The model weight of one perceptron is independent of the other; therefore, any relationship within the input data is lost. CNN models treat inputs as tensors, allowing the CNN filters (learning units) to look for specific patterns anywhere in the data. In this study, the 33 pressure sensors are spatially related to each other with respect to leak scenarios. In most instances, spatially similar leak scenarios (leak scenarios with similar leak locations) tend to induce leak signatures at some common set of pressure sensors. CNN models can learn this spatial relationship, while MLP models cannot. This difference in spatial awareness of the two models is evident in the higher difference in performance between the CNN and the MLP model for the *random leaks* case, which has a higher number of spatially similar leak scenarios compared to the other three cases.

The second reason for the superior performance of CNN models is related to their ability to handle noise in the data. As shown in studies by Lan et al. (2021) and Rhodin and Kvist (2019), models that are based on CNN architecture are better at reducing noise in the data. Consequently, the CNN model performs better for the *demand-noise* and *mixed-noise* cases. Finally, due to the fully connected nature of MLP models and their need for an additional perceptron for each additional input data, the number of parameters of MLP models increases rapidly (LeCun et al. 1998, 1990), forming a redundant and inefficient web structure. This structure is difficult to train and suffers from overfitting. CNN models do not face this problem because they treat the input as tensors, which enables parameter sharing (LeCun et al. 1998, 1990; Krizhevsky et al. 2017). In this study, when the input increases from a single time point to 24 time points, the number of MLP parameters increases by 329% compared to a mere 1% for the CNN model. Such a high increase in the number of parameters makes the MLP model susceptible to overfitting. Therefore, at the lowest threshold (0.1), where overfitting has a high impact on the accuracies, the CNN accuracies are vastly superior to the MLP accuracies.

Single Time Input versus Multiple Time Input

In real-world systems, the pressure sensors are programmed to record pressure readings at a particular time interval, which are transferred back to the central control unit or SCADA. The leak localization algorithm, therefore, has a choice to use one or multiple time point pressure data. The general assumption is that multiple time point pressure data contains more leak information than single time point data, which can improve leak localization accuracies. This assumption is tested by comparing the precision and recall for a single time point pressure input and a 24 time points (multiple time) pressure input for the four study cases (with both CNN and MLP).

Little-to-no improvements in precision and recall are observed for the *no-noise* study case with the 24 time points pressure data (Fig. 13). Because the single time point data for the *no-noise* case are noiseless pressure readings recorded at the 33 pressure nodes 1 h after the leaks are introduced, there is ample unaltered leak signature in the first-hour pressure data itself, which is evident by the high precision and recall values with single time point data. Therefore, incorporating additional time points data does not improve the precision or recall further. Similarly, the accuracy of the *demand-noise* case (Fig. 14) is almost unaltered when additional time point data are incorporated. Because the noise added to the demand parameters of the hydraulic model prior to simulation induces comparatively small resultant noise in the pressure readings, the single point data carries as much information as the 24 time point data for the *demand-noise* case.

In contrast to the first two cases, considerable improvements (>10%) in the precision and recall are observed for the *mixed-noise* case with the 24 time point data with both models (Fig. 15). Even though 10% gaussian noise is added to the pressure data, the noisy pressure data are still distributed around the actual pressure signal. A better approximation of the actual pressure signal is possible with 24 time points compared to a single time point. These approximate pressure signals can contain sufficient information, especially when the actual signatures are prominent. Therefore, machine learning models can learn better with 24 time points compared to a single time point. Hence, the accuracies improve when multiple time points data are incorporated. For the *random leaks* case (Fig. 16), precision and recall improve for both MLP and CNN models; however, the improvement is comparatively less (<4%). As previously discussed, the complexity of the *random leaks* case is due to the ambiguity caused by the possibility of a multitude (all unique) of leak signatures (pressure signals) for multiple leak scenarios considered the same (leaks in the same candidate region but located differently within the candidate region). This ambiguity would be removed if the pressure signals for the multiple leak scenarios (considered the same) shared similarity at most of the available 24 time points. Even though there exists a possibility that a few of the time points may share some similarity, the chances that most

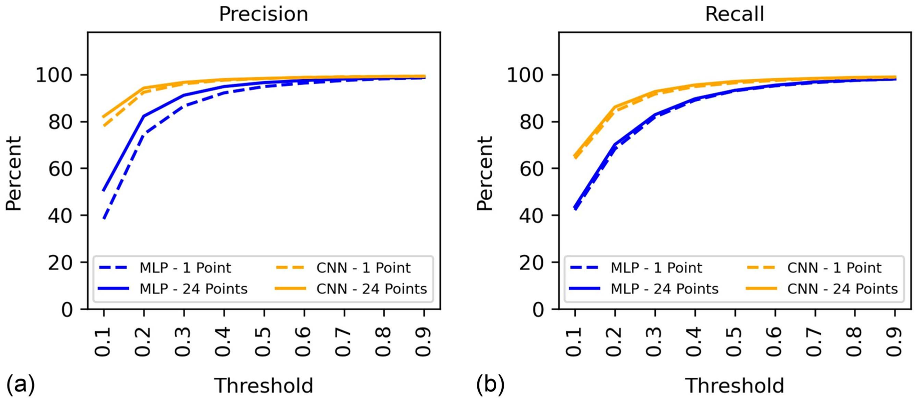

Fig. 13. Comparative model performance (single versus 24 time points) for no-noise case: (a) precision; and (b) recall.

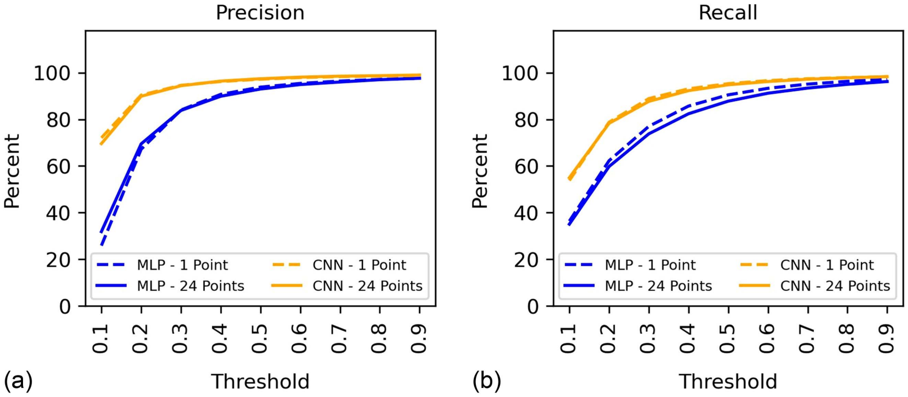

Fig. 14. Comparative model performance (single versus 24 time points) for demand-noise case: (a) precision; and (b) recall.

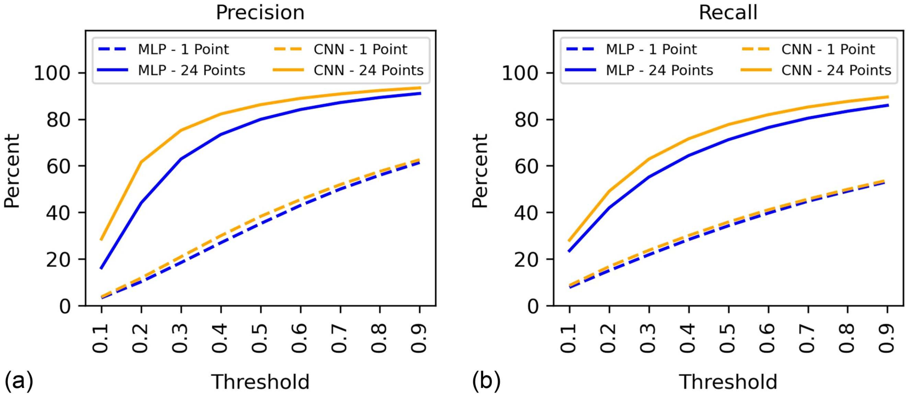

Fig. 15. Comparative model performance (single versus 24 time points) for mixed-noise case: (a) precision; and (b) recall.

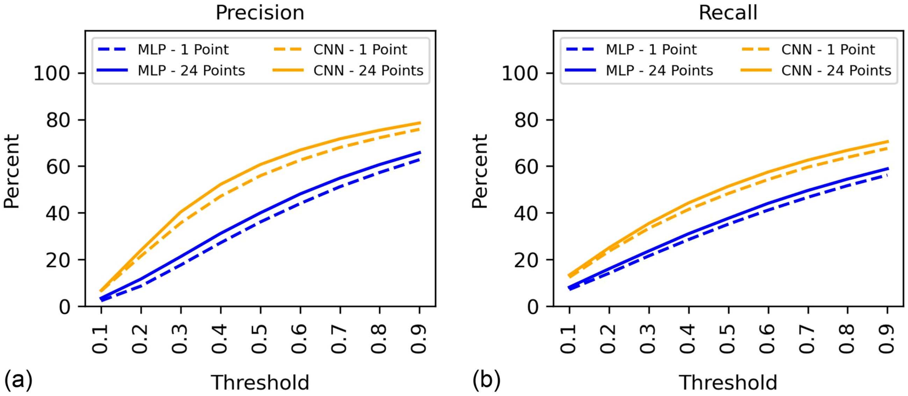

Fig. 16. Comparative model performance (single versus 24 time points) for random leaks case: (a) precision; and (b) recall.

or all time points will be the same are rare. Therefore, using 24 time points pressure data induces minimal improvement in the accuracies, if any.

Single Input (Pressure Data) versus Multiple Input (Pressure and Flow Data)

Precision and recall for three selected study cases (with both CNN and MLP) are compared to understand the potential of using multiple types of inputs simultaneously for leak localization. The three study cases are *no-noise*, *mixed-noise*, and *random leaks*. The results for the *demand-noise* case are highly similar to the *no-noise* as observed in the previous sections; therefore, they are not presented.

Flow data are fed simultaneously along with the pressure data to train the models and localize the leaks. The flow data consists of the flow from the ten sensors recorded for the first hour after the leaks. Improvements in both precision and recall at one or all thresholds are observed for all study cases with both CNN and MLP models (Figs. 17–19). The critical difference between the *no-noise* and the

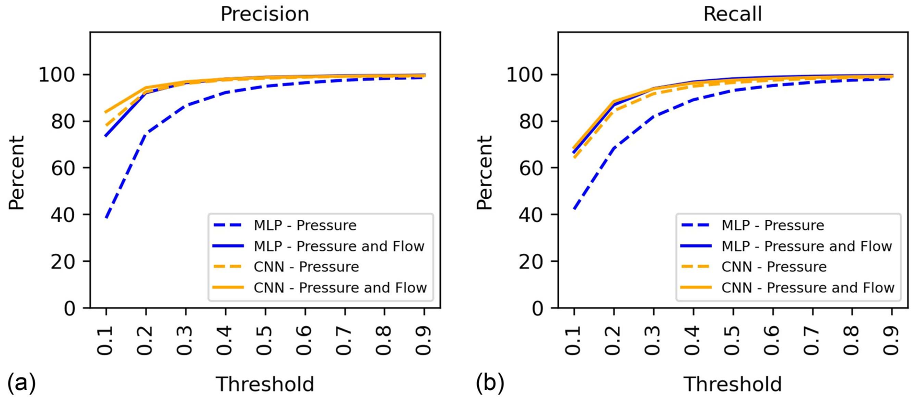

Fig. 17. Comparative model performance (pressure versus pressure and flow) for no-noise case: (a) precision; and (b) recall.

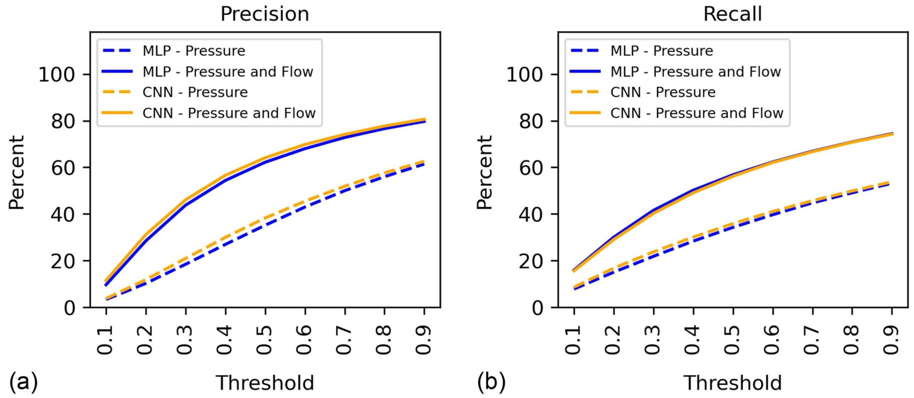

Fig. 18. Comparative model performance (pressure versus pressure and flow) for mixed-noise case: (a) precision; and (b) recall.

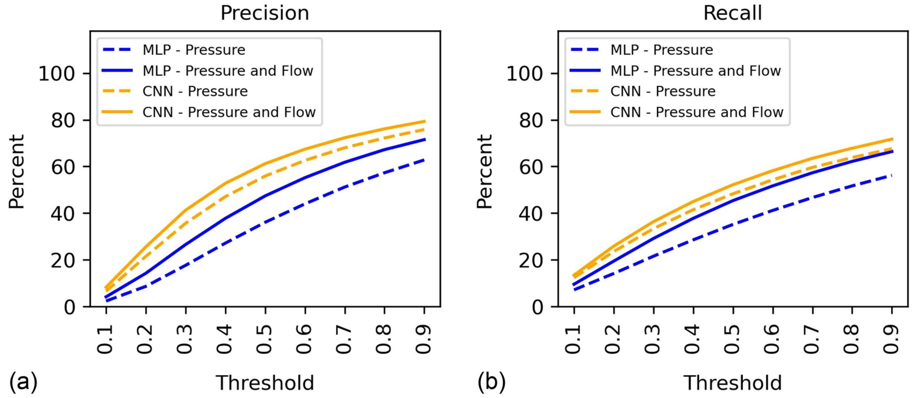

Fig. 19. Comparative model performance (pressure versus pressure and flow) for random leaks case: (a) precision; and (b) recall.

other cases is the thresholds where the improvements occur due to the additional flow data. For the *no-noise* case, considerably large (>25%) improvements are observed at the lower thresholds and slight improvements (<2%) are observed at the higher thresholds. For the remaining cases, the improvements are low (<10%) at the lower and high (>20%) at the higher thresholds. This difference between the *no-noise* and the other cases can be attributed to two main reasons. First, the precision and recall at the higher thresholds for the *no-noise* case are very high (>80%) compared to the lower thresholds (<45%), leaving less room for improvement at the higher thresholds. The other reason is associated with the data quality and the problem complexity. The *no-noise* case, as previously discussed, is an ideal simplistic problem assisted by the lack of noise in any available data. Therefore, depending upon the quality of leak information in the flow data, additional flow input can proportionally improve the leak localization performance. The *mixed-noise* and *random leaks* cases are challenging due to noise and ambiguity in the data. Even with additional flow input data, these challenges still exist, and affect the model accuracies. The impact of this noise and ambiguity is especially more at the lower thresholds where we are trying to pinpoint the size of the leaks (in addition to the location) within such low tolerances. With pressure data only, the precision and recall for the *mixed-noise* and *random leaks* cases are less than 10% at the 0.1 thresholds (Figs. 18 and 19). With additional flow data, minor improvements, less than 10% for *mixed-noise* and less than 5% for *random leaks*, are observed at the 0.1 thresholds. Considerably high improvements (>20%) are observed at higher thresholds as the problem comparatively simplifies.

Similar results are observed for CNN and MLP models. Precision and recall improve for both models due to additional flow data, but the amount of improvement is more for the MLP model compared to the CNN model. The CNN model, however, still outperforms the MLP model for all cases with the additional flow input.

Summary and Conclusions

In this study, a machine learning–based approach is proposed for localizing leaks in WDSs that evaluates the impact of the different complexities of leak localization problems. The impact of WDS leak characteristics (varying size, multiple occurrences, and random location) and the uncertainties associated with the hydraulic model parameter and measuring devices, the possible benefit of multiple time points data, and the potential of multiple different simultaneous input data are studied by analyzing the performance of two different readily available and widely used machine learning models. The results of this study highlight the necessity of considering the various characteristics of real-world leaks in creating leak scenarios to properly understand the applicability and effectiveness of the leak localization models. Simplistic and unrealistic leak scenarios, such as the *no-noise* case, overestimate the performance of the models, as seen in this study. Models trained with only such scenarios can severely underperform and be deemed useless for real WDSs. However, the high accuracies of the CNN and the MLP models trained and tested with the three realistic study cases involving data noise, random leaks, and model and instrument uncertainties prove their potential for application to real-world leak localization problems. This study also found that the effectiveness of the machine learning–based leak localization method is model-dependent; the level of challenge posed by the same complexity of the leak localization problem affects the performance of varying machine learning models differently. In this study, the CNN model is more effective than the MLP model in localizing leaks. The CNN model's superior architecture, input handling, and learning ability enable it to learn the spatial relationship in the input data, better tackle noise in the data, and share parameters to produce more generalizable results than the MLP model. Future researchers can take guidance from these results when faced with the task of selecting machine learning models for their leak localization studies.

It is also important to point out that the locations of the pressure sensors used to generate the input data in this study are not based on hydraulic analysis and, therefore, are not optimal. Optimally located pressure nodes can further improve the accuracies of the models. The results from this study also concur with the general understanding that additional time point data improves leak localization accuracies. However, the improvements are not always comparable or substantial due to noisy data and complex leak characteristics. Finally, this study shows that it is feasible to simultaneously use multiple types of input data to improve the localization of leaks. However, ensuring that these inputs are reliable in terms of quality and leak information is necessary. One potential limitation of any simulated data–based approaches, including this study, is associated with the calibration conditions of the hydraulic model representing the actual WDS. For cases where the calibration is conducted in the presence of background leaks in the system, the resulting models can characterize new leaks only.

Several possibilities remain open for extending this study further. First, various other types of noise can be considered in the input data to account for any additional real-world leak complexity. Next, the findings of the simultaneous use of pressure and flow data also open the possibility of using other types of information, such as acoustics, water quality, and more. Ultimately, the goal is to use these models in practice. Therefore, the final step will test these leak detection models to real-world WDSs to truly understand their potential.

Data Availability Statement

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request, including pressure, flow, and leak response data for the four cases presented in this study.

Acknowledgments

This material is based upon work supported by the National Science Foundation (NSF) under Partnership for Innovation (PFI) Grant No. 1919228.

References

Abadi, M., et al. 2016. "Tensor flow: Large-scale machine learning on heterogeneous distributed systems." Accessed December 13, 2022. https://arxiv.org/abs/1603.04467.

Abdulla, M. B., and R. Herzallah. 2015. "Probabilistic multiple model neural network based leak detection system: Experimental study." *J. Loss Prev. Process Ind.* 36 (Jul): 30–38. https://doi.org/10.1016/j.jlp.2015.05.009.

Berglund, A., V. S. Areti, D. Brill, and G. K. Mahinthakumar. 2017. "Successive linear approximation methods for leak detection in water distribution systems." *J. Water Resour. Plann. Manage.* 143 (8): 1–13. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000784.

Blesa, J., and R. Pérez. 2018. "Modelling uncertainty for leak localization in water networks." *IFAC-Pap. Online* 51 (24): 730–735. https://doi.org/10.1016/j.ifacol.2018.09.656.

Bohorquez, J., B. Alexander, A. R. Simpson, and M. F. Lambert. 2020. "Leak detection and topology identification in pipelines using fluid transients and artificial neural networks." *J. Water Resour. Plann. Manage.* 146 (6): 04020040. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001187.

Eliades, D. G., M. Kyriakou, S. Vrachimis, and M. M. Polycarpou. 2016. *EPANET-MATLAB toolkit: An open-source software for interfacing EPANET with MATLAB*. Amsterdam, Netherlands: Computer Control for Water Industry. https://doi.org/10.5281/zenodo.437751.

Fontanazza, C. M., V. Notaro, V. Puleo, P. Nicolosi, and G. Freni. 2015. "Contaminant intrusion through leaks in water distribution system: Experimental analysis." *Procedia Eng.* 119 (1): 426–433. https://doi.org/10.1016/j.proeng.2015.08.904.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep learning*. Cambridge, MA: MIT Press.

Hu, Z., D. Tan, B. Chen, W. Chen, and D. Shen. 2021. "Review of model-based and data-driven approaches for leak detection and location in water distribution systems." *Water Supply* 21 (7): 3282–3306. https://doi.org/10.2166/ws.2021.101.

Kingma, D. P., and J. L. Ba. 2014. "Adam: A method for stochastic optimization." Preprint, submitted December 13, 2022. https://arxiv.org/abs/1412.6980.

Klise, K. A., M. Bynum, D. Moriarty, and R. Murray. 2017. "A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study." *Environ. Modell. Software* 95 (Sep): 420–431. https://doi.org/10.1016/j.envsoft.2017.06.022.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2017. "ImageNet classification with deep convolutional neural networks." *Commun. ACM* 60 (6): 84–90. https://doi.org/10.1145/3065386.

Lan, R., H. Zou, C. Pang, Y. Zhong, Z. Liu, and X. Luo. 2021. "Image denoising via deep residual convolutional neural networks." *Signal Image Video Process.* 15 (1): 1–8. https://doi.org/10.1007/s11760-019-01537-x.

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1990. "Handwritten digit recognition with a back-propagation network." In *Advances in neural information processing systems*, 2. Denver: Morgan Kaufmann.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. "Gradient-based learning applied to document recognition." *Proc. IEEE* 86 (11): 2278–2324. https://doi.org/10.1109/5.726791.

Liemberger, R., and A. Wyatt. 2019. "Quantifying the global non-revenue water problem." *Water Sci. Technol. Water Supply* 19 (3): 831–837. https://doi.org/10.2166/ws.2018.129.

Lloyd, S. P. 1982. "Least squares quantization in PCM." *IEEE Trans. Inf. Theory* 28 (2): 129–137. https://doi.org/10.1109/TIT.1982.1056489.

Lučin, I., Z. Carija, S. Druzeta, and B. Lučin. 2021. "Detailed leak localization in water distribution networks using random forest classifier and pipe segmentation." *IEEE Access* 9 (Nov): 155113–155122. https://doi.org/10.1109/ACCESS.2021.3129703.

Maas, A. L., and A. Y. Ng. 2013. "Rectifier nonlinearities improve neural network acoustic models." In *Proc., 30th Int. Conf. on Machine Learning*, 28. Stanford, CA: Stanford Univ.

Mashford, J., D. D. Silva, S. Burn, and D. Marney. 2012. "Leak detection in simulated water pipe networks using SVM." *Appl. Artif. Intell.* 26 (5): 429–444. https://doi.org/10.1080/08839514.2012.670974.

Mohammed, E. G., E. B. Zeleke, and S. L. Abebe. 2021. "Water leakage detection and localization using hydraulic modeling and classification." *J. Hydroinf.* 23 (4): 782–794. https://doi.org/10.2166/hydro.2021.164.

Moser, G., S. German Paal, and I. F. C. Smith. 2018. "Leak detection of water supply networks using error-domain model falsification." *J. Comput. Civ. Eng.* 32 (2): 04017077. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000729.

Pérez-Pérez, E. J., F. R. López-Estrada, G. Valencia-Palomo, L. Torres, V. Puig, and J. D. Mina-Antonio. 2021. "Leak diagnosis in pipelines using a combined artificial neural network approach." *Control Eng. Pract.* 107 (Feb): 104677. https://doi.org/10.1016/j.conengprac.2020.104677.

Ponce, M. V. C., L. E. G. Castañón, and V. P. Cayuela. 2014. "Model-based leak detection and location in water distribution networks considering an extended-horizon analysis of pressure sensitivities." *J. Hydroinf.* 16 (3): 649–670. https://doi.org/10.2166/hydro.2013.019.

Rajeswaran, A., N. Sridharakumar, and S. Narasimhan. 2018. "A graph partitioning algorithm for leak detection in water distribution networks." *Comput. Chem. Eng.* 108 (Jan): 11–23. https://doi.org/10.1016/j.compchemeng.2017.08.007.

Rhodin, S. L., and E. Kvist. 2019. *A comparative study between MLP and CNN for noise reduction on images: The impact of different input dataset sizes and the impact of different types of noise on performance*. Stockholm, Sweden: KTH Royal Institute of Technology.

Santos-Ruiz, I., J. Blesa, V. Puig, and F. R. López-Estrada. 2020. "Leak localization in water distribution networks using classifiers with cosenoidal features." *IFAC-Pap. Online* 53 (2): 16697–16702. https://doi.org/10.1016/j.ifacol.2020.12.1113.

Shukla, H., and K. Piratla. 2020. "Leakage detection in water pipelines using supervised classification of acceleration signals." *Autom. Constr.* 117 (Sep): 103256. https://doi.org/10.1016/j.autcon.2020.103256.

Soldevila, A., J. Blesa, S. Tornil-Sin, E. Duviella, R. M. Fernandez-Canti, and V. Puig. 2016. "Leak localization in water distribution networks using a mixed model-based/data-driven approach." *Control Eng. Pract.* 55 (Oct): 162–173. https://doi.org/10.1016/j.conengprac.2016.07.006.

Soldevila, A., R. M. Fernandez-Canti, J. Blesa, S. Tornil-Sin, and V. Puig. 2017. "Leak localization in water distribution networks using Bayesian classifiers." *J. Process Control* 55 (Jul): 1–9. https://doi.org/10.1016/j.jprocont.2017.03.015.

Steffelbauer, D. B., and D. Fuchs-Hanusch. 2016. "Efficient sensor placement for leak localization considering uncertainties." *Water Resour. Manage.* 30 (14): 5517–5533. https://doi.org/10.1007/s11269-016-1504-6.

Vrachimis, S. G., D. G. Eliades, R. Taormina, A. Ostfeld, Z. Kapelan, S. Liu, M. Kyriakou, P. Pavlou, M. Qiu, and M. M. Polycarpou. 2020. "Dataset of BattLeDIM: Battle of the leakage detection and isolation methods." Zenodo. Accessed September 7, 2020. https://doi.org/10.5281/zenodo.4017659.

Wu, Y., and S. Liu. 2017. "A review of data-driven approaches for burst detection in water distribution systems." *Urban Water J.* 14 (9): 972–983. https://doi.org/10.1080/1573062X.2017.1279191.

Wu, Z. Y., P. Sage, and D. Turtle. 2010. "Pressure-dependent leak detection model and its application to a district water system." *J. Water Resour. Plann. Manage.* 136 (1): 116–128. https://doi.org/10.1061/(ASCE)0733-9496(2010)136:1(116).

Yu, J., L. Zhang, J. Chen, Y. Xiao, D. Hou, P. Huang, G. Zhang, and H. Zhang. 2021. "An integrated bottom-up approach for leak detection in water distribution networks based on assessing parameters of water balance model." *Water* 13 (6): 867. https://doi.org/10.3390/w13060867.

Zaman, D., M. K. Tiwari, A. K. Gupta, and D. Sen. 2020. "A review of leakage detection strategies for pressurised pipeline in steady-state." *Eng. Fail. Anal.* 109 (Jan): 104264. https://doi.org/10.1016/j.engfailanal.2019.104264.

Zhang, Q., Z. Y. Wu, M. Zhao, J. Qi, Y. Huang, and H. Zhao. 2016. "Leakage zone identification in large-scale water distribution systems using multiclass support vector machines." *J. Water Resour. Plann. Manage.* 142 (11): 1–15. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000661.

Zhou, M., Z. Pan, Y. Liu, Q. Zhang, Y. Cai, and H. Pan. 2019a. "Leak detection and location based on ISLMD and CNN in a pipeline." *IEEE Access* 7: 30457–30464. https://doi.org/10.1109/ACCESS.2019.2902711.

Zhou, X., Z. Tang, W. Xu, F. Meng, X. Chu, K. Xin, and G. Fu. 2019b. "Deep learning identifies accurate burst locations in water distribution networks." *Water Res.* 166 (Dec): 115058. https://doi.org/10.1016/j.watres.2019.115058.