doi: https://doi.org/10.4995/WDSA-CCWI2022.2022.14831

SUPERVISED MACHINE LEARNING MODELS FOR LEAK DETECTION IN WATER DISTRIBUTION SYSTEMS

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

[1,2,3,4]Department of Civil Engineering, North Carolina State University, Raleigh, NC, (USA)

[1]lbasnet@ncsu.edu, [2]ranji@ncsu.edu, [3]brill@ncsu.edu, [4]gmkumar@ncsu.edu

Abstract

Water distribution systems (WDSs) face a significant challenge in the form of pipe leaks. Pipe leaks can cause loss of a large amount of treated water, leading to pressure loss, increased energy costs, and contamination risks. Locating pipe leaks has been a constant challenge for water utilities and stakeholders due to the underground location of the pipes. Physical methods to detect leaks are expensive, intrusive, and heavily localized. Computational approaches provide an economical alternative to physical methods. Data-driven machine learning-based computational approaches have garnered growing interest in recent years to address the challenge of detecting pipe leaks in WDSs. While several studies have applied machine learning models for leak detection on single pipes and small test networks, their applicability to the real-world WDSs is unclear. Most of these studies simplify the leak characteristics and ignore modeling and measuring device uncertainties, which makes the scalability of their approaches questionable to real-world WDSs. Our study addresses this issue by devising four study cases that account for the realistic leak characteristics (multiple, multi-size, and randomly located leaks) and incorporating noise in the input data to account for the model- and measuring device- related uncertainties. A machine learning-based approach that uses simulated pressure as input to predict both location and size of leaks is proposed. Two different machine learning models: Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN), are trained and tested for the four study cases, and their performances are compared. The precision and recall results for the L-Town network indicate good accuracies for both the models for all study cases, with CNN generally outperforming MLP.

Keywords
Leak detection, machine learning, multilayer perceptron, convolutional neural network, hydraulic simulation, water distribution systems.

1 INTRODUCTION

Water distribution systems (WDSs) face a significant challenge in the form of pipe leaks. Pipe leaks can cause loss of large amount of treated water in WDSs leading to pressure loss and increased energy costs. Leaks can also pose risks of water contamination [1]. As reported in [2], an estimated 126 billion cubic meters of water is lost every year worldwide. With increasing demands and growing concerns about water scarcity in the face of climate change, the prevention of water losses from WDSs is crucial. Moreover, pipe leaks can grow over time and lead to breaks and bursts causing property damage and traffic disruptions. Therefore, timely detection and prevention of pipe leaks are paramount. Unlike pipe breaks, pipe leaks are tough to detect as the flow or pressure changes produced by leaks are not humanly discernable [3]. In addition, the underground location of pipes makes it even harder to detect leaks. Physical methods to detect leaks are expensive and can cause interruption to water service [4]. Computational approaches provide an economical alternative to physical methods.

Several computational approaches have been proposed for leak detection (an extensive review is provided in [5]). Machine learning approaches are one of the data-driven computational approaches that have gathered increasing interest in the last two decades in leak detection studies [6]. Machine learning methods use a large amount of data related to the hydraulic properties of WDSs such as pressure, flowrate, acoustic vibration, optics, or temperature for leak detection [7]. Pressure and flowrates are the most commonly used properties for leak detection [8].

While a good amount of research has been conducted on the application of machine learning models for leak detection in pipes [9], the question about their applicability to real-world WDSs remains unclear. One of the critical reasons for this lack of clarity concerns the scalability of the approaches considered in these studies. In [10] and [11], Convolutional Neural Network (CNN) was used for leak detection in a single pipe using simulated negative pressure wave and scalogram images of vibration signals as inputs, respectively. In [12], MLP was used 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 a viable solution in the field as 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. MLP was used in [13] to predict leaks in a simple hydraulic system using numerically obtained fluid transient waves as input. In [14], SVM was used to predict leak size and location for an isolated section of a WDS based on simulated pressure data. In [15], a model-based k-Nearest Neighbors (k-NN) classifier was used to identify leak events and locations. These studies still face the challenge of scalability as extrapolating their results and, therefore, application to the larger real-world WDSs is very challenging.

Another factor that limits the real-world application of some of the existing machine learning approaches relates to the simplifying assumptions regarding the characteristics of pipe leaks in WDSs. For example, the application of the Bayesian classifier in [16] to detect leaks assumes that there is only a single leak in the WDS, which is rarely true. In [17], unsupervised principal component analysis (PCA) was used for leak detection by assuming a single, constant size leak.

Furthermore, very few studies have considered uncertainties associated with hydraulic simulation models and imprecision of measurement devices in real-world WDSs. The parameters such as demands, pipe roughness, pipe diameters, and lengths used in the hydraulic models have associated uncertainties [18]. These uncertainties affect the accuracy of the simulated pressure and flow data. One way to account for the hydraulic model parameter uncertainties is to add noise to these parameters prior to simulation, as shown in [19] and [20]. However, such an approach is inadequate to encapsulate the uncertainties related to the imprecision of measurement devices such as pressure sensors and flow meters of the real-world WDSs.

This study proposes a machine learning-based approach to detecting and localizing leaks in WDSs, which considers multiple realistic leak scenarios and accounts for hydraulic model uncertainties and instrument imprecision. Two different machine learning models are used to predict leaks using simulated pressure measurements as input. The key contributions of this study with respect to previous approaches include:

- Overcoming the unrealistic simplification about occurrence of a single leak at a time assumed by most state-of-the-art techniques [15] by generalizing to multi-leak 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, i.e., demand uncertainty, as well as measuring instrument imprecision through the addition of noise to the input data.
- Simultaneous prediction of location as well as siz e of the leaks.

Even though the machine learning models are trained using simulated pressure data, they are applicable to predict leaks using real-world measurements as long as the hydraulic model is a reasonable representation of the real system. For WDSs that have abundant real-world pressure sensor measurements, these models can easily be fine-tuned and tested using the real data.

2 METHODS

2.1 General Framework

Figure 1 illustrates the general framework proposed in this study to detect and localize leaks in WDS pipes. The framework starts with a WDS hydraulic model that generates simulated operational pressure data. First, pressure data for a leak-free scenario is generated by simulating the hydraulic model using the EPANET simulator [21]. It is followed by pressure data generation for multiple different leak scenarios. Pressure differences between the leak scenarios and the leak-free scenario are then computed and stored as a pressure readings dataset. The corresponding leak scenarios are stored as a leak values dataset. Noise is then added to the pressure readings dataset when required for the case under study described in Section 2.5. The resulting pressure readings dataset and the leak values dataset are then randomly shuffled and split into training sets and testing sets; a train to test ratio of 80 to 20 is used. The training pressure dataset and the training leak values dataset are scaled and fed to machine learning models. The pressure data is considered as covariates and the leak values as responses. The models are trained and tuned, and the optimized models are selected for the prediction of leaks. Finally, the leak prediction and model evaluation are performed on the testing pressure and leak values datasets using the optimized models; predicted model outputs are compared with the corresponding true leak values.

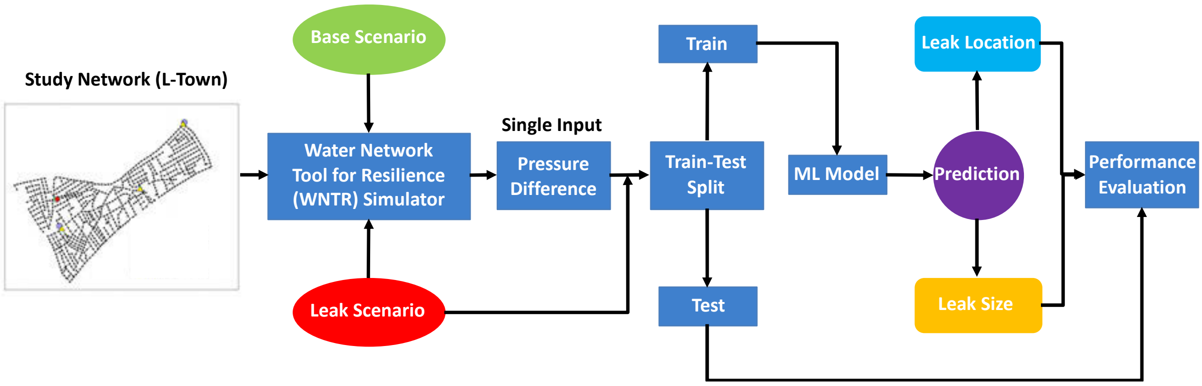


Figure 1. General framework for detecting leaks in WDS

2.2 Machine Learning Models

2.2.1 Multilayer Perceptron (MLP)

Multilayer Perceptrons (MLPs) are supervised-learning models based on deep neural networks. An MLP model consists of an input layer, an output layer, and a selected number of dense hidden layers located between the input and the output layers (Figure 2). Activation layers follow hidden layers to activate or deactivate received signals. Multiple activation functions are available to be used in these activation layers.

2.2.2 1-D Convolutional Neural Network (CNN)

2.2.2 1-D Convolutional Neural Network (CNN)

Similar to MLPs, Convolutional Neural Networks (CNNs) are also supervised-learning-based deep neural networks. The key difference between CNNs and MLPs is the presence of convolutional and pooling layers in CNNs. As shown in Figure 3, the convolutional layers produce convolved feature maps, which allow for contextual learning, and the pooling layers downsample these maps to extract abstract features from the data. The convolutional layers use kernels or filters to extract the features. A one-dimensional (1-D) CNN model uses filters that only vary in depth (i.e., one dimension). CNN models also have an input, an output layer, some dense hidden layers, and activation layers similar to the MLPs.

2.2.3 Hyperparameters and Model Tuning

Total Number of Iterations (Epochs): MLP and CNN models are trained for a number of iterations (epochs) to ensure the stability in the training process. The optimal model and its corresponding weights are determined by monitoring the training and validation errors over the entire number of epochs.

Error Function: The functions to calculate the training and validation errors are chosen based on the nature of the problem. In this study, leak detection is formulated as a regression type problem to simultaneously solve for both leak locations and sizes. Therefore, the mean squared error (MSE) function is used; mean absolute error (MAE) can be used as an alternative to MSE.

Activation Function: A trial-and-error evaluation of multiple activation functions identified the Leaky Rectified Linear Unit (L-ReLU) as a suitable activation function for this study. L-ReLU prevents the problem of vanishing gradient during forward propagation like the regular rectified linear unit (ReLU) and has an added advantage of preventing vanishing gradients during backward propagation [22].

Optimizer: The commonly used Adam optimizer is used in this study.

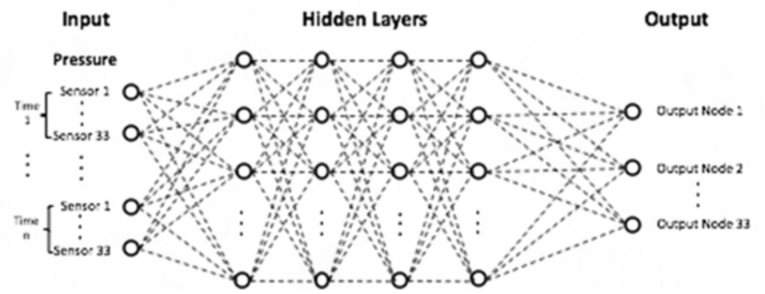


Figure 2: Multilayer Perceptron

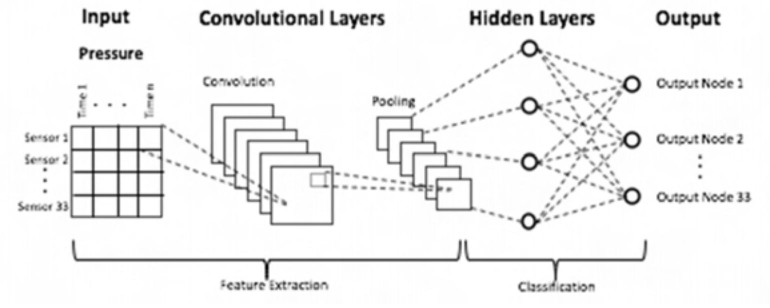


Figure 3: Convolutional Neural Network

2.3 Study Network

In this study, the leak detection methods are applied to a standard test network called the L-Town water network (Figure 4). The L-Town network has been previously used in several modeling and simulation related researches. For example, this network was also used in the Leakage Detection and Isolation Methods (BattLeDIM 2020) [23] 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.

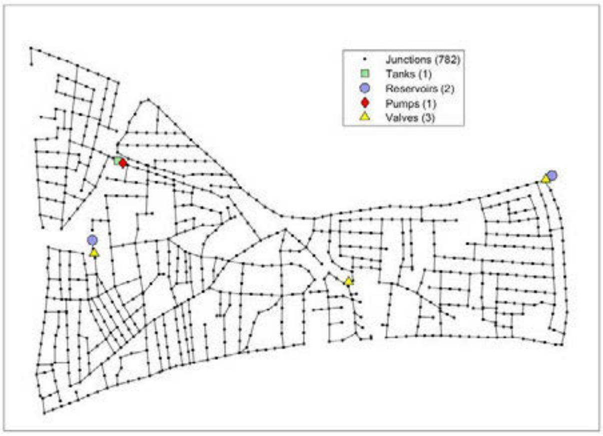
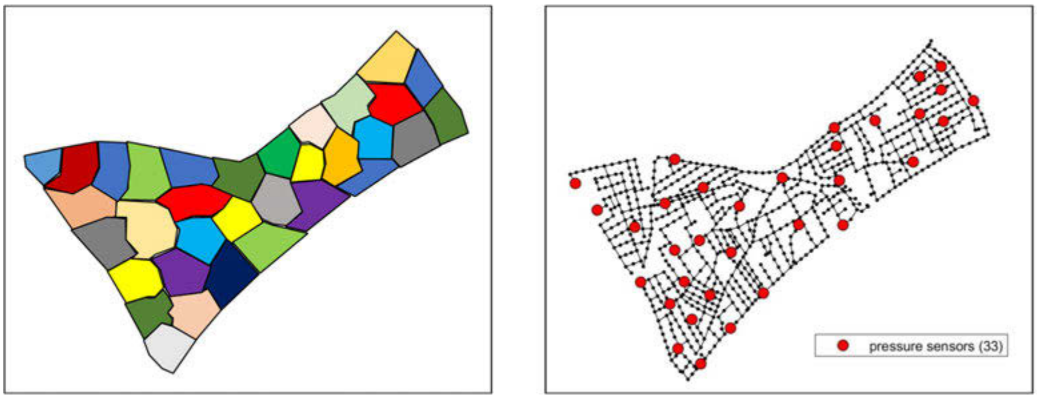


Figure 4: L-Town water network

2.4 Candidate Leak Regions

Localizing leaks to the actual pipes or junctions requires large amount of data, which is infeasible to obtain from real-world WDSs. Therefore, a lesser resolution is adopted for leak localization in this study. The entire water network is divided into several sub-areas that are considered candidate leak regions. The L-Town network is divided into 33 candidate leak regions (Figure 5a). A k-means clustering technique [24] is used to divide the network into these 33 candidate leak regions based on Euclidean distances. Leaks are modelled as emitters in EPANET and are assumed to occur at the center of each pipe. Since EPANET supports emitters only on nodes, new junction nodes are inserted at the middle of every pipe in the network using the Morph package in WNTR [25]. Candidate leak nodes representing each leak region is assumed to be at the centroid of each leak region. Centroids of leak regions are estimated using k-nearest neighbour search algorithm. For any given leak scenario, a leak located anywhere within the boundaries of a candidate leak region is defined by this region. While a hydraulic distance-based clustering measure results in more homogeneous clusters, the less homogeneous clusters obtained using Euclidean distance-based measure may pose a more significant challenge for the leak detection models. Therefore, the Euclidean distance-based clustering used here is a more conservative approach.

A pressure node is assigned to each of the 33 candidate regions to track the pressure changes due to leak/s 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 Figure 5b.



(a) (b)

Figure 5: Candidate leak regions: (a) Leak regions; (b) Pressure sensors within each region

2.5 Study Cases

Four study cases are considered in this study to represent the realistic leak characteristics, and the uncertainties in input data due to water network model inaccuracies and measuring device imprecision are considered for this study.

Case A: No-noise – Input pressure difference data is free of noise. It represents the ideal case of accurate WDS models and precise measuring devices. Leaks are assumed to occur at the centroid of each leak region.

Case B: Demand-noise – Input pressure difference data accounts for the WDS model inaccuracies. To mimic the inaccuracies in demand values in the WDS model, random Gaussian noise are added to the demands prior to simulation. Simulated pressure data are then generated using the modified WDS model. A ten percent Gaussian noise is used. Leaks are assumed to occur at the centroid of each leak region.

Case C: Mixed-noise – Input pressure difference data accounts for the WDS model inaccuracies as well as the measuring device imprecision. Unlike the *demand-noise* case, noise is added to the final pressure differences of the leak and leak free scenarios. A ten percent Gaussian noise is used. Leaks are assumed to occur at the centroid of each leak region.

Case D: Random leaks – The leaks can be located anywhere within the boundaries of the candidate leak regions instead of their centroids. No additional noise is imposed.

2.6 Data Generation

The input datasets used in this study constitute the leak scenario and the pressure difference datasets, which are generated in two sequential steps.

2.6.1 Leak Scenario Generation

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

- A leak scenario must consist of at least one 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 coefficient in the leak equation (1).

$$q = C\,p^{\Upsilon} \tag{1}$$

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

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

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

Step 2 – Based on the outcome ***n*** of the previous draw, ***n*** candidate leak locations out of the 33 candidate leak locations are drawn at random.

Step 3 – For these ***n*** candidate leak locations, the leak sizes are randomly drawn from the leak size range of 0 to 5.

Step 4 – Repeat steps 1 – 3 for 100,000 times to generate 100,000 leak scenarios.

The 100,000 leak scenarios generated from the above procedure were saved as a leak scenario dataset.

2.6.2 Pressure Data Generation

Simulated pressure data are generated by the following procedure:

Step 1 – As discussed in Section 2.4, assign one pressure node each to all 33 candidate regions. The locations of the pressure nodes are based on the locations used in BattLeDIM 2020.

Step 2 – Simulate a leak-free scenario for the specified study case defined in Section 2.5 by running the base model with the EPANET simulator. Store the resulting pressure values at the 33 pressure nodes.

Step 3 – Pick a leak scenario from the leak scenario dataset and add the associated leaks to the base model. Then, run this modified model with the EPANET simulator and store the resulting pressure values at the 33 pressure nodes.

Step 4 – Repeat Step 3 for all the 100,000 leak scenarios in the dataset.

Step 5 – Compute the pressure differences between each of the 100,000 leak scenarios and the leak-free scenario. Then, combine the 100,000 pressure differences together as a pressure difference dataset.

Step 6 – Add noise to the pressure difference data depending upon the study case discussed in Section 2.5.

2.7 Model Validation and Testing

2.7.1 Train-Test Split

The input pressure and leak datasets are divided into training and test data. A training to test ratio of 80 to 20 is used to split the data. The two models are validated using the test datasets.

2.7.2 Metrics and Thresholds

The performance of the two machine learning models to predict leaks is evaluated using the two standard classification metrics: precision and recall.

$$Precision = \frac{TP}{TP + FP} \times 100 \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \times 100 \tag{3}$$

where, TP = True Positives; FP = False Positives; and FN = False Negatives.

In the context of this study, precision is the percentage of the actual leaks out of all leak predictions made by the models. Recall is the percentage of the actual leaks identified by our models out of all the leaks in the dataset.

In this study, the problem of leak detection is formulated as a regression type problem to simultaneously solve for both leak locations and sizes. To assess the model performances in terms of precision and recall, a post-processing of model outputs is required. This post-processing involves the use of 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 that are smaller than 0.1 in the dataset are considered as no-leaks and only the predictions that are within 0.1 units of the actual leak values are considered as correct classifications.

2.8 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 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.

3 RESULTS AND DISCUSSION

Leak prediction performance of the MLP and CNN models are studied for the four study cases described in Section 2.5. The two models are compared by calculating precision and recall accuracies for the test dataset. Table 1 summarizes the architecture and hyperparameters for the optimal MLP and CNN models. The optimal MLP model has four dense layers: the input layer and the output layer, with 33 units each, and the two central dense layers with 64 and 128 units. The optimal CNN model consists of six layers - four dense layers and two convolutional layers. Like MLP, two out of the four dense layers are the input and the output layers, with 33 units each. The remaining two dense layers are hidden layers with 500 and 100 units, respectively. The two convolutional layers (also hidden) that follow the input layers consist of 256 and 128 filters. Figure 6 shows the trend of the training and validation mean squared errors for the *no-noise* case for the two models. The validation errors show a general decreasing trend that stops after the 100th epoch, indicating 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 is set to 100 epochs.

Table 1. Machine learning model details

Model	Architecture	Hidden Layers	Dense Layers	Convolutional Layers	Activation Functions	Learning Rate	Optimizer
MLP	33-64-128-33	2	4	-	LReLU	0.05	Adam
CNN	33-256-128-500-100-33	4	4	2	LReLU	0.05	Adam

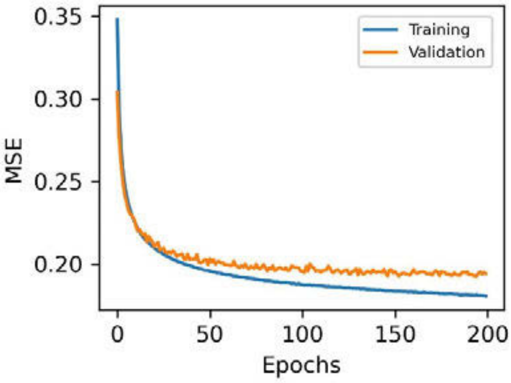
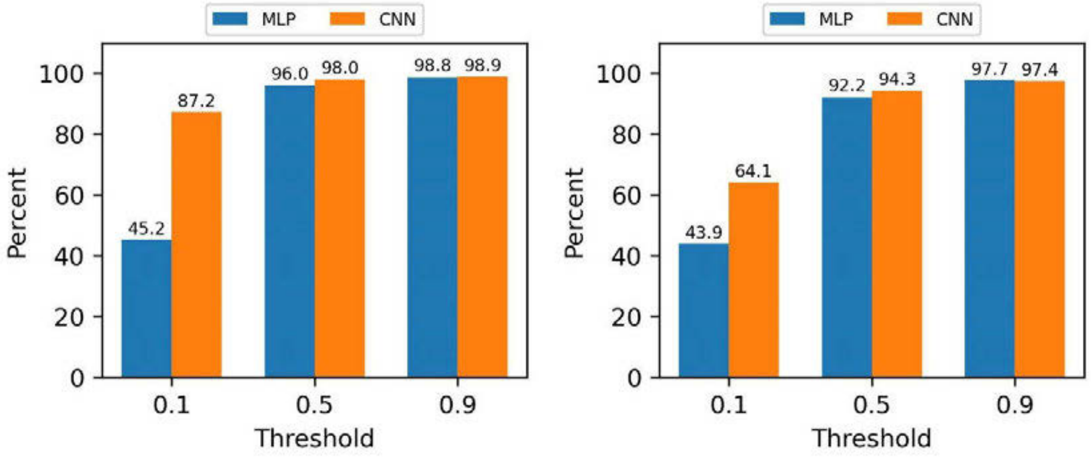


Figure 6. Training and validation error dynamics

3.1 Comparison of CNN and MLP model performance

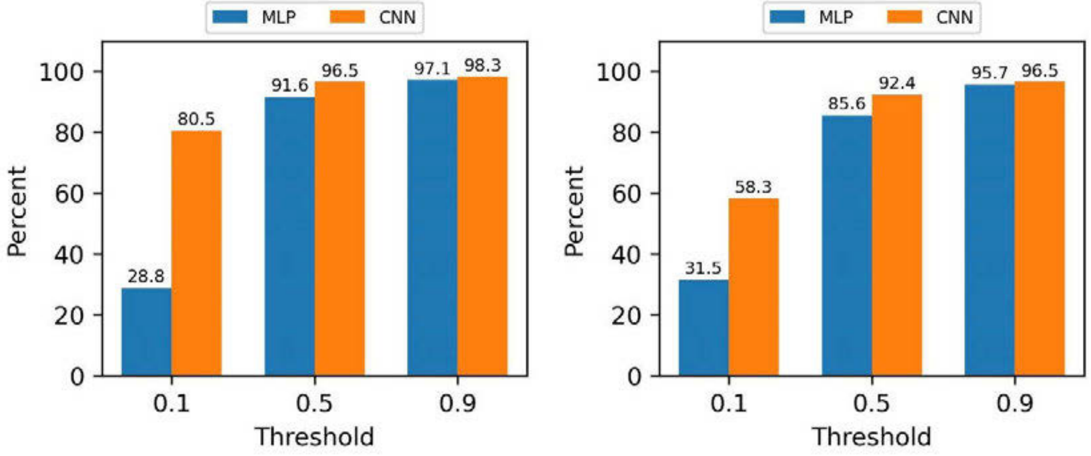
The complete model performances for the MLP and the CNN models for the four study cases are summarized in Tables 2 and 3. Figures 7 – 10 show the precision and recall for the two models at three selected thresholds (0.1, 0.5, and 0.9) for the four study cases (no-noise, demand-noise, mixed-noise, and random leaks). The results at these three thresholds are representative of all nine thresholds considered in this study, with 0.1, 0.5, and 0.9 indicating the most, the mild, and the least stringent condition, respectively. The figures show that precision is generally high (> 60%) for the CNN model at all three thresholds for all study cases except for the random leak case. Comparatively, precision for the MLP model is lower at all thresholds. The difference in precision between the two models is significantly high (> 40%) at 0.1 threshold for the no-noise, demand-noise, and mixed-noise cases. This difference, however, starts to diminish as the threshold becomes less stringent. 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 even with noise in the input data.

Similar to precision, recall for the CNN model is higher than the MLP model at the most stringent threshold (0.1) for all four study cases. However, the difference in recall of the two models at 0.1 threshold is not as high as the difference in precision. At the lesser stringent thresholds, particularly at 0.9, the difference in recall for the two models is insignificant for the no-noise, demand-noise, and mixed-noise cases. However, this difference is significant for the random leak case at all thresholds, with the CNN model outperforming the MLP model throughout. Overall, the recall results are consistent with the precision results in implying the superior performance of the CNN model over the MLP for the L-Town network.



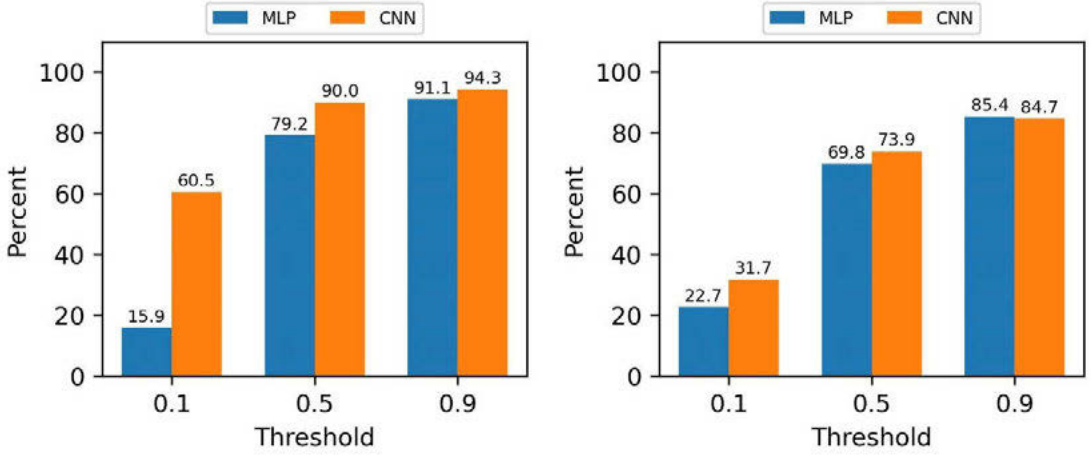
(a) *(b)*

Figure 7. Model performance for no-noise case: (a) Precision; (b) Recall.



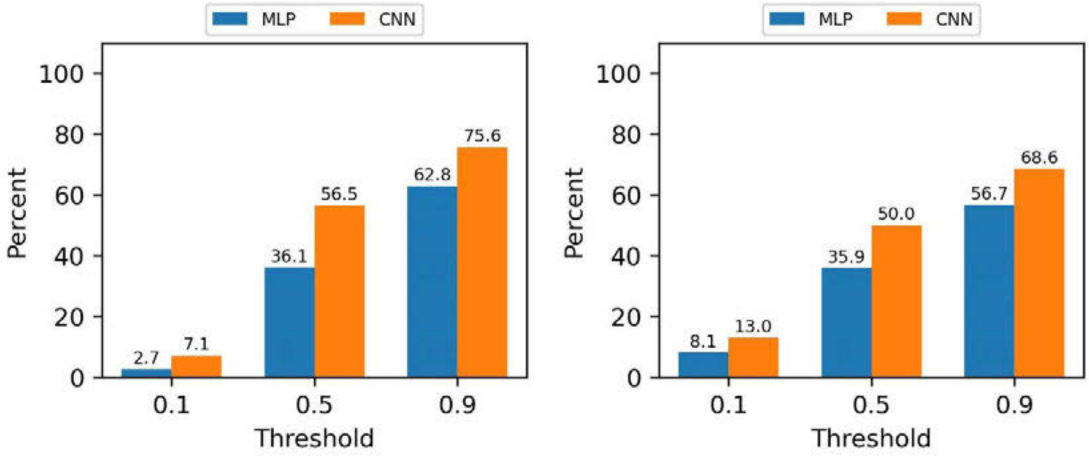
(a) *(b)*

Figure 8. Model performance for demand-noise case: (a) Precision; (b) Recall.



(a) *(b)*

Figure 9. Model performance for mixed-noise case: (a) Precision; (b) Recall.



(a) *(b)*

Figure 10. Model performance for random leaks case: (a) Precision; (b) Recall.

3.2 Problem complexity of the study cases

The precision and recall for the four study cases (with both CNN and MLP) were compared to understand the complexity of the leak detection task associated with each of the cases. Both precision and recall at all thresholds for the ideal but unrealistic *no-noise* case (Figure 7) rank highest compared to the other three study cases (Figure 8 – 10) 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 without noise. The *demand-noise* case ranks second among these four cases based on the 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 systematic. Therefore, it affects the leak signatures to a lesser degree. For the *mixed-noise* case, the 10% Gaussian noise added to the input pressure differences introduces randomness in the data that affect the leak signatures to a comparatively greater degree. Therefore, the precision and recall of the two models for the *mixed-noise* case are significantly lower compared to *no-noise* and *demand-noise* cases. The leak signatures are affected to the highest degree for the *random leaks* case. While no direct noises are added to the input data as is done for the *mixed-noise* case, the randomness in leak locations within a leak region introduces the possibility of a multitude of leak signatures for the same leak scenario, which is the most challenging for the machine learning models to learn. Therefore, the *random leaks* case ranks lowest in precision and recall. The effect of the complexity of the *mixed-noise* and the *random leaks* cases is profound at the 0.1 threshold because the artificial noise created by the randomness in the input data drowns out the changes in pressure input caused by a leak size or leak size difference of 0.1.

Table 2. Precision for the study cases

Threshold	No-noise		Demand-noise		Mixed-noise		Random leaks	
	MLP	CNN	MLP	CNN	MLP	CNN	MLP	CNN
0.1	45.2	87.2	28.8	80.5	15.9	60.5	2.7	7.1
0.2	78.5	94.4	63.7	89.7	43.2	77.0	9.1	22.5
0.3	89.4	96.3	79.9	93.5	61.6	84.0	17.7	36.9
0.4	93.9	97.4	87.4	95.4	72.6	87.7	27.1	48.0
0.5	96.0	98.0	91.6	96.5	79.2	90.0	36.1	56.5
0.6	97.3	98.5	94.0	97.2	83.7	91.5	44.2	63.0
0.7	98.0	98.7	95.5	97.7	87.0	92.6	51.2	68.3
0.8	98.5	98.8	96.5	98.1	89.3	93.5	57.3	72.3
0.9	98.8	98.9	97.1	98.3	91.1	94.3	62.8	75.6

Table 3. Recall for the study cases

Threshold	No-noise		Demand-noise		Mixed-noise		Random leaks	
	MLP	CNN	MLP	CNN	MLP	CNN	MLP	CNN
0.1	43.9	64.1	31.5	58.3	22.7	31.7	8.1	13.0
0.2	69.3	82.5	54.9	77.2	40.5	50.4	15.3	24.8
0.3	81.6	88.9	69.6	85.4	53.0	62.0	22.1	34.9
0.4	88.4	92.3	79.2	89.7	62.7	69.0	29.2	43.1
0.5	92.2	94.3	85.6	92.4	69.8	73.9	35.9	50.0
0.6	94.6	95.6	89.8	94.0	75.1	77.7	41.8	55.8
0.7	96.1	96.4	92.6	95.1	79.3	80.6	47.3	60.8
0.8	97.0	97.0	94.4	95.8	82.6	82.8	52.2	65.0
0.9	97.7	97.4	95.7	96.5	85.4	84.7	56.7	68.6

4 CONCLUSIONS

In this study, a machine learning-based approach is proposed for detecting leaks in WDSs that takes into account the characteristics of leaks present in real-world WDSs. 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 are studied by analyzing the performance of two different machine learning models. One of the key findings of this study is that the effectiveness of the machine learning-based leak detection method is model-dependent. In this study, the CNN model is more effective than the MLP model in detecting leaks. While this result is specific to the study network (L-Town) using pressure differences as input, its implication expands beyond this study. It establishes the need to explore multiple models when developing a leak detection method. The other key finding of this study highlights the necessity of considering various types of leak scenarios that bear realistic leak characteristics to understand better the applicability of the leak detection models to real WDSs. Simplistic and unrealistic leak scenarios such as the *no-noise* case overestimate the performance of the models, as seen in this study. Models trained under such scenarios can severely underperform and be deemed useless for real WDSs. However, the high accuracies of the CNN and the MLP models for the three realistic study cases involving data noise, random leaks, and model and instrument uncertainties are proof of their potential for application to real-world leak detection problems. 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.

Several possibilities remain open for improving the work done in this study. Continuing the exploration of real-world leak characteristics, the addition of other types of noise can be considered for the input data. Using multiple inputs instead of a single input such as pressure is another possibility to improve leak detection accuracy. Our work in progress includes adding flow data alongside pressure data to predict leak locations and size. Finally, to understand the true potential of these leak detection models, the next step forward for this study is to apply them to a real-world WDS.

5 REFERENCES

[1] C. M. Fontanazza, V. Notaro, V. Puleo, P. Nicolosi, and G. Freni, “Contaminant intrusion through leaks in water distribution system: Experimental analysis,” Procedia Eng., vol. 119, no. 1, pp. 426–433, 2015.

[2] R. Liemberger and A. Wyatt, “Quantifying the global non-revenue water problem,” Water Sci. Technol. Water Supply, vol. 19, no. 3, pp. 831–837, 2019.

[3] J. Yu et al., “An integrated bottom-up approach for leak detection in water distribution networks based on assessing parameters of water balance model,” Water (Switzerland), vol. 13, no. 6, 2021.

[4] A. Rajeswaran, S. Narasimhan, and S. Narasimhan, “A graph partitioning algorithm for leak detection in water distribution networks,” Comput. Chem. Eng., vol. 108, pp. 11–23, 2018.

[5] Z. Hu, D. Tan, B. Chen, W. Chen, and D. Shen, “Review of model-based and data-driven approaches for leak detection and location in water distribution systems,” Water Supply, vol. 21, no. 7, pp. 3282–3306, 2021.

[6] D. Zaman, M. K. Tiwari, A. K. Gupta, and D. Sen, “A review of leakage detection strategies for pressurised pipeline in steady-state,” Eng. Fail. Anal., vol. 109, no. October 2019, 2020.

[7] I. Santos-Ruiz, J. Blesa, V. Puig, and F. R. López-Estrada, “Leak localization in water distribution networks using classifiers with cosenoidal features,” IFAC-PapersOnLine, vol. 53, no. 2, pp. 16697–16702, 2020.

[8] M. B. Abdulla and R. Herzallah, “Probabilistic multiple model neural network based leak detection system: Experimental study,” J. Loss Prev. Process Ind., vol. 36, pp. 30–38, 2015.

[9] Y. Wu and S. Liu, “A review of data-driven approaches for burst detection in water distribution systems,” Urban Water J., vol. 14, no. 9, pp. 972–983, 2017.

[10] M. Zhou, Z. Pan, Y. Liu, Q. Zhang, Y. Cai, and H. Pan, “Leak Detection and Location Based on ISLMD and CNN in a Pipeline,” IEEE Access, vol. 7, pp. 30457–30464, 2019.

[11] H. Shukla and K. Piratla, “Leakage detection in water pipelines using supervised classification of acceleration signals,” Autom. Constr., vol. 117, no. May, p. 103256, 2020.

[12] E. J. Pérez-Pérez, F. R. López-Estrada, G. Valencia-Palomo, L. Torres, V. Puig, and J. D. Mina-Antonio, “Leak diagnosis in pipelines using a combined artificial neural network approach,” Control Eng. Pract., vol. 107, no. September 2020, p. 104677, 2021.

[13] J. Bohorquez, B. Alexander, A. R. Simpson, and M. F. Lambert, “Leak Detection and Topology Identification in Pipelines Using Fluid Transients and Artificial Neural Networks,” J. Water Resour. Plan. Manag., vol. 146, no. 6, p. 04020040, 2020.

[14] J. Mashford, D. De Silva, S. Burn, and D. Marney, “Leak detection in simulated water pipe networks using SVM,” Appl. Artif. Intell., vol. 26, no. 5, pp. 429–444, 2012.

[15] A. Soldevila, J. Blesa, S. Tornil-Sin, E. Duviella, R. M. Fernandez-Canti, and V. Puig, “Leak localization in water distribution networks using a mixed model-based/data-driven approach,” Control Eng. Pract., vol. 55, pp. 162–173, 2016.

[16] A. Soldevila, R. M. Fernandez-Canti, J. Blesa, S. Tornil-Sin, and V. Puig, “Leak localization in water distribution networks using Bayesian classifiers,” J. Process Control, vol. 55, pp. 1–9, 2017.

[17] M. Quinõnes-Grueiro, C. Verde, A. Prieto-Moreno, and O. Llanes-Santiago, “An unsupervised approach to leak detection and location in water distribution networks,” Int. J. Appl. Math. Comput. Sci., vol. 28, no. 2, pp. 283–295, 2018.

[18] J. Blesa and R. Pérez, “Modelling uncertainty for leak localization in Water Networks ˜ ,” vol. 51, no. 24, pp. 730–735, 2018.

[19] X. Zhou et al., “Deep learning identifies accurate burst locations in water distribution networks,” Water Res., vol. 166, p. 115058, 2019.

[20] E. G. Mohammed, E. B. Zeleke, and S. L. Abebe, “Water leakage detection and localization using hydraulic modeling and classification,” J. Hydroinformatics, vol. 23, no. 4, pp. 782–794, 2021.

[21] D. G. Eliades, M. Kyriakou, S. Vrachimis, and M. M. Polycarpou, “EPANET-MATLAB toolkit: An open-source software for interfacing EPANET with MATLAB,” in 14th Int. Conf. on Computing and Control for the Water Industry, 2016.

[22] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[23] S. G. Vrachimis et al., “BattLeDIM: Battle of the leakage detection and isolation methods.,” in Proc., 2nd Int. CCWI/WDSA Joint Conf., 2020.

[24] S. P. Lloyd, "Least Squares Quantization in PCM," IEEE Trans. Inf. Theory, vol. 28, no. 2, pp. 129–137, 1982.

[25] K. A. Klise, M. Bynum, D. Moriarty, and R. Murray, "A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study," Environ. Model. Softw., vol. 95, pp. 420–431, 2017.