Enhanced Water Leak Detection with Convolutional Neural Networks and One-Class Support Vector Machine

Daniele Ugo Leonzio
Politecnico di Milano
Milan, Italy
danieleugo.leonzio@polimi.it

Paolo Bestagini
Politecnico di Milano
Milan, Italy
paolo.bestagini@polimi.it

Marco Marcon
Politecnico di Milano
Milan, Italy
marco.marcon@polimi.it

Stefano Tubaro
Politecnico di Milano
Milan, Italy
stefano.tubaro@polimi.it

***Abstract*—Water is a critical resource that must be managed efficiently. However, a substantial amount of water is lost each year due to leaks in Water Distribution Networks (WDNs). This underscores the need for reliable and effective leak detection and localization systems. In recent years, various solutions have been proposed, with data-driven approaches gaining increasing attention due to their superior performance. In this paper, we propose a new method for leak detection. The method is based on water pressure measurements acquired at a series of nodes of a WDN. Our technique is a fully data-driven solution that makes only use of the knowledge of the WDN topology, and a series of pressure data acquisitions obtained in absence of leaks. The proposed solution is based on an feature extractor and a one-class Support Vector Machines (SVM) trained on no-leak data, so that leaks are detected as anomalies. The results achieved on a simulate dataset using the Modena WDN demonstrate that the proposed solution outperforms recent methods for leak detection.**

***Index Terms*—Anomaly detection, water leak detection**

I. Introduction

Between the growing threats of climate change and pollution, the shift to an environmentally sustainable economy is gaining momentum. This transition highlights the need to understand the environmental impact of our actions. Key strategies include transforming energy production, reducing food waste, and achieving carbon negativity, all aimed at cost-effective sustainability with a positive environmental footprint.

A notable trend is the increased monitoring of critical infrastructure to ensure better working conditions, reduce waste, and boost ecological benefits. Monitoring helps anticipate issues, implement preventive maintenance, and optimize production, enhancing operational efficiency and environmental contributions.

In WDNs, a major concern is water leakage. Factors such as rising demand, aging infrastructure, and environmental degradation have worsened water scarcity in recent years. Leaks, which account for about 30% of urban water usage, have significant economic and environmental impacts [1]. Rapid and reliable leak detection in pipe networks offers considerable advantages, reducing operational costs, improving service levels for water utilities, and preventing water pollution. Detecting leaks is not just an economic issue but also an environmental and safety imperative.

In response to these challenges, modern electronic and information technologies have enabled the development of advanced sensing systems for water leak detection [2]–[7].

To address the limitations of hardware-based approaches, data-driven methods have gained prominence in recent years. These approaches leverage statistical analysis and machine learning algorithms to identify and locate leaks within WDNs without the need for detailed pipe-specific information [8]. Data-driven solutions prioritize the utilization of broad network topology information to pinpoint the location of leaks. They effectively bypass the requirement for detailed pipe parameters such as diameter, material composition, and thickness. For instance, data-driven methods such as SVM have been employed to solve the leak detection problem based on pressure sensor measurements and transient data [2], [9]. Essentially, these approaches learn from historical data using statistical or pattern recognition tools [10], endowing them with a robust generalization capability. Generalization allows them to be effectively applied across a diverse range of network configurations, regardless of the specific details of individual pipes. In simpler terms, a data-driven solution developed for one particular WDN can, with its inherent adaptability, be successfully implemented in other networks with entirely different pipe characteristics.

In recent years, deep learning approaches have gained traction due to their effectiveness in leak detection. These techniques have revolutionized the field by providing advanced frameworks for handling complex pat-

ISBN 978-3-903176-72-0 © 2025 IFIP

terns and variabilities. Quiñones-Grueiro et al. [11] integrated deep neural networks with Gaussian process regression to detect and localize leaks, demonstrating the potential of combining data-driven and model-based approaches. The Burst Location Identification Framework (BLIFF) framework utilizes pressure data in a mixed data-driven and model-based approach, a method less explored in other studies to pinpoint leaks more accurately by focusing on specific pipes rather than general areas. To address the complexity of pressure signals, linear connections were used in place of the typical convolutional layers in DenseNet [12], simplifying calculations. The work [13] by Geelen et al. explores the use of pressure sensor data to enhance the monitoring and management of water distribution systems. The authors develop a framework, called Monitoring Support, that utilizes real-time pressure data to detect anomalies and improve the system's overall reliability. The detection is done using a moving window range statistic and the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) method [14].

Soldevilla et al. [15] propose a method that detects leaks by a sequential monitoring algorithm that analyzes the inlet flow of a District Metered Area (DMA). A DMA is a portion of a more complex WDN. This sectorization is done through valves or disconnection of network pipes with inlet and outlet flow metered [16]. They formulated the leak detection problem as a change-point detection problem, solved by an ad hoc two layer algorithm including a hypothesis test to validate each detection and estimate the leak size and leak time.

A novel deep learning approach is proposed in [17]. In this work the authors propose to solve the leak detection and localization problem using an autoencoder trained on leak free samples. While the authors achieved improvements in Detection Delay (DD) and localization error, their work is limited to a small network that is not representative of a real-world WDN.

In this work we propose a new method to detect leak in a WDN. The proposed method analyzes measurements of water pressure at a series of nodes of a WDN. In particular, we train a feature extractor on top of measurements obtained in case of no leaks, as done in [17], [18]. After with the feature extracted from the leak free samples we train a one-class SVM. At test time, the one-class SVM detects leaks as anomalies. The results achieved are promising, as we obtain an average DD of about 40.21 hours (i.e., we detect a leak after 40.21 hours). This improvement was demonstrated on the Modena WDN, a real-size network, underscoring the practical applicability of the proposed approach.

The paper is organized as follows. Section II reports the formal description of the problem addressed in this work and describes the main steps of the proposed technique. Section III focuses on presenting the dataset, the autoencoder training, and how we select the parameters of our method. Section IV presents the performance results on leak detection. Finally, Section V concludes the paper.

II. Water Leak Detection

Problem Formulation. In our work we propose a method for water leak detection in a WDN by monitoring the pressures values measured at a series of nodes.

Let us consider a WDN composed by K nodes. Each node is equipped with a pressure sensor. The pressure measured at the k-th node is a time series sampled at regular intervals. The n-th sample at the k-th node is defined as $p_k(n)$.

The presence of a leak modifies the pressure p_k with a drop of the pressure value at the sample $N^\star$ corresponding to the leak starting time $T^\star$, with a reduction that is proportional to the leak size.

With this setup in mind, we can define the two main goals of this paper as follows

- *To detect if a leak starts in a WDN*: this means attributing to the WDN under analysis a label $\hat{c}$ equals to 1 if a leak is detected or 0 otherwise.
- *To determine the leak starting time*: this means to compute $\hat{T}$ which is an estimate of $T^\star$.

Proposed Method. We propose solving the leak detection problem using an algorithm based on a feature extractor and a one-class SVM. The feature extractor is derived from the encoder section of an autoencoder, a specialized type of neural network designed to encode input into a compressed and meaningful representation, then decode it back so that the reconstructed input closely resembles the original [19]. The autoencoder is trained on data representing a no-leak scenario, allowing the system to encode the input pressure matrix into a compact vector. By training with no-leak samples, we ensure that anomalies are not introduced during the training phase. The idea of using an autoencoder for leak detection in a WDN trained on no-leak samples has already been explored in the literature [17]. In this work, we use the autoencoder proposed in [17] as our starting point. The one-class SVM is trained on the embeddings extracted from the no-leak samples, in order to spot the embeddings from leak scenarios as anomalies. The feature extraction step is essential because training a one-class SVM directly on the raw input data can be less effective due to the high dimensionality and complexity of the data. The encoder compresses and refines the input into a more meaningful and compact representation, making it easier for the one-class SVM to accurately detect anomalies.

As shown in Fig. 1, our pipeline can be broken down into 3 main blocks: *Preprocessing*; *Feature Extraction*; *Detection*.

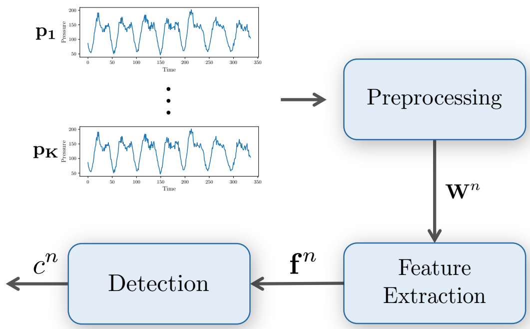

Fig. 1. Block diagram of the proposed method.

In the following, we report all the details related to each one of these blocks.

1) Preprocessing: Preprocessing is a crucial step for preparing the data before it is input into the feature extractor. During this stage, the system receives a new sample $p_k(n)$ from each node simultaneously at every time instant. The preprocessing component collects the most recent L samples from all K sensors and arranges them into a fixed-size matrix $K \times L$, which will then be processed by the feature extractor in the next phase.

This component operates as a circular buffer, continuously updating by discarding the oldest sample whenever a new one is obtained. This method ensures that the analysis is consistently performed on a fixed-length window, concentrating solely on the most recent data.

At each sampling instant n, the preprocessing step selects a window of L samples from p_k, comprising the most recent $L-1$ samples along with the latest one.

Since data from all nodes are processed simultaneously, the matrix $\mathbf{W}^n$ is formed by concatenating the last L pressure samples collected from all nodes within the WDN, as shown:

$$\mathbf{W}^n = \begin{bmatrix} p_1(n-L+1) & \dots & p_1(n-1) & p_1(n) \\ p_2(n-L+1) & \dots & p_2(n-1) & p_2(n) \\ \vdots & \vdots & \vdots & \vdots \\ p_K(n-L+1) & \dots & p_K(n-1) & p_K(n) \end{bmatrix}, \quad (1)$$

where the k-th row represents measurements from the k-th node, and K is the number of nodes in the WDN.

2) Feature Extraction: The second step of our method consists in analyzing the input matrix $\mathbf{W}^n$ with a feature extractor in order to have a compact and meaningful representation of it. The feature extractor is obtained as the encoder part of an autoencoder. The autoencoder is trained to reconstruct the input signal in case of absence of leak. In this way, the training process of the autoencoder is not affected by possible anomalies in the input matrix $\mathbf{W}^n$. The complete description of the autoencoder and its training process can be found in Section III.

The feature extractor takes the input $\mathbf{W}^n$ with size $K \times L$, and computes a reduced dimensionality version of it. This is composed by 2 convolutional layers:

- The first layer is a 1D Convolutional Layer with 64 filters of size 7 and stride 2, followed by a Rectified Linear Unit (ReLU).
- The second layer is a 1D Convolutional Layer with 32 filters of size 7 and stride 2, followed by a ReLU.

The output of the feature extractor model is vector $\mathbf{f}^n$, which represent the reduce dimensionality version of the input matrix $\mathbf{W}^n$.

$$\mathbf{f}^n = \mathcal{E}(\mathbf{W}^n). \quad (2)$$

3) Detection: The detection step of the proposed method is based on a one-class SVM. The one-class SVM approach is a machine learning method used for anomaly detection in scenarios where the training data consists solely of instances from one class, typically representing normal behavior. This model learns the boundary that best encompasses the normal data, mapping it into a high-dimensional feature space where it defines a hyperplane or a boundary around the majority of the data points. During testing, the one-class SVM classifies new instances based on their proximity to this boundary: data points that lie within the boundary are classified as normal, while those that fall outside are flagged as anomalies or outliers [20]. This approach is especially useful in scenarios where anomalous data is rare or challenging to collect for training, such as in the case of water leak detection.

The one-class SVM has been trained using the no-leak scenario embeddings from the same samples used to train the autoencoder.

The output of the one-class SVM is a soft score $\hat{y}^n$.

To analyze the behavior over time, we concatenate all the predicted values $\hat{y}$ for each analyzed window, forming a score vector denoted as $\hat{\mathbf{y}}$. To suppress noise and patterns unrelated to anomalies, we apply a moving average filter to the vector $\hat{\mathbf{y}}$ obtaining the vector $\hat{\mathbf{y}}_{\text{smoothed}}$. Finally, we apply a threshold γ to the smoothed vector, designating all values exceeding this threshold as anomalies.

Formally,

$$\hat{c}^n = \begin{cases} 1, & \text{if } \hat{y}^n_{\text{smoothed}} \geq \gamma, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Here, $\hat{c}^n$ represents the binary anomaly indicator at the sample n, where values in the smoothed vector $\hat{\mathbf{y}}_{\text{smoothed}}$ above the threshold γ are set to 1.

If a leak is detected (i.e., $\hat{c}^n = 1$ for at least one value of n), the time instant associated to the last received

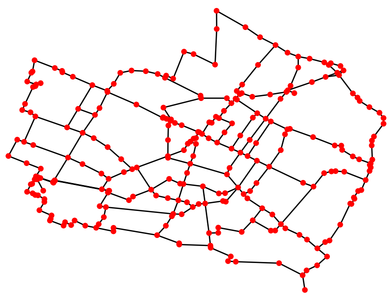

Fig. 2. Modena network topology. Each dot represents a node. Lines represent connections between nodes.

pressure sample is set as leak starting time instant. We denote the position of this sample as $\hat{N}$, which corresponds to the time instant $\hat{T}$.

III. Experimental Setup

Dataset. To develop and test our algorithm, we utilized the Modena WDN, a network frequently employed as a benchmark in various studies [21]–[23]. The network comprises 268 nodes and 317 pipes, with no pumps or valves included in the model. Each pipeline in the network is characterized by its diameter, length, and roughness. The pipe lengths range from 1 m to 1094.73 m, while diameters vary between 100 mm and 400 mm.

We simulated 500 distinct scenarios, including both leak and no-leak cases. In the leak scenarios, we varied the leak positions and sizes, while all scenarios shared the same fixed network topology. The simulations were conducted using EPANET [24]. To further increase the dataset's complexity, we introduced randomness to the diameter, length, and roughness values of each pipe.

In Fig. 2 we report the topology of the Modena network used in our work.

In each scenario, we have a simulation of a WDN sampled every 30 minutes for one year, so every 30 minutes a new sample is added and the buffer is shifted. The node number K is equal to 268 for all scenarios. For every node we have the pressure time series.

Autoencoder. The autoencoder can be divided in two parts: the *Encoder* ($\mathcal{E}$) and the *Decoder* ($\mathcal{D}$).

The encoder takes the input $\mathbf{W}^n$ with size $K \times L$, and computes a reduced dimensionality version of it. The encoder layers are described in Section II-2 The decoder takes the low-dimensionality output of the encoder as input, and estimates the output $\hat{\mathbf{W}}^n$ with size $K \times L$. This is composed by 3 convolutional layers:

- The first layer is a 1D Transposed Convolutional Layer with 32 filters of size 7 and stride 2, followed by a ReLU.
- The second layer is a 1D Transposed Convolutional Layer with 64 filters of size 7 and stride 2, followed by a ReLU.
- The third layer is a 1D Transposed Convolutional Layer with 32 filters of size 7 and stride 1.

Both the encoder and the decoder parameters have been selected after a grid search procedure, and the structure proposed here is the best in terms of loss value and computational cost.

We train the autoencoder on no-leak data considering Mean Squared Error (MSE) as loss function. The output of the autoencoder is $\hat{\mathbf{W}}^n$, a reconstruction of the input matrix $\mathbf{W}^n$ obtained as

$$\hat{\mathbf{W}}^n = \mathcal{AE}(\mathbf{W}^n), \tag{4}$$

where $\mathcal{AE}$ implements the autoencoder operator. The Eq. (4) can be written also as

$$\hat{\mathbf{W}}^n = \mathcal{D}(\mathcal{E}(\mathbf{W}^n)), \tag{5}$$

where the autoencoder model $\mathcal{AE}$ has been split in to its encoder $\mathcal{E}$ and decoder part $\mathcal{D}$.

Parameters selection. In this section we report additional details about the parameters we adopt in our pipeline.

The first parameter is the sample length L of the window. We set L in order to cover one week of measurements within each sliding window. This means that $L = 336$ samples considering the simulation sampling rate. Thanks to this choice we are able to model both daily and weekly periodicity, while still remaining robust to seasonal variations because these are slower than the window size and window update. This parameter set automatically also the setup time of the algorithm, which is equal to the window length (i.e., one week). The setup time refers to the initial period required for the algorithm to start functioning effectively, during which it accumulates enough data to populate the first window.

The second parameter is the threshold γ reported in Section II-3. The γ value has been chosen comparing the results obtained with different values of False Positive Rate (FPR), on a validation set formed by 50 simulations. In our method we decide to adopt a FPR of 10% which gave as a threshold value of 7.44.

IV. Results

This section presents the results we achieved with our method on Modena network dataset. We compare our detection results against [17] and a refined version of [17], in which we change the original threshold proposed by the authors in order to adapt the method on the different data used in this work. In addition we test the performance of the proposed method in case of noisy data with different Signal to Noise Ratio (SNR) levels and the generalization capabilities on a different WDN.

Detection. The performance of the leak detection system was evaluated using accuracy as the primary metric.

TABLE I
DETECTION ACCURACY

Method	**Accuracy**
Baseline [17]	0.22
Baseline [17] refined	0.80
Proposed Method	0.92

TABLE II
DETECTION DELAY IN HOURS

Method	**Detection Delay**
Baseline [17]	71.00
Baseline [17] refined	61.64
Proposed Method	40.21

A true positive is defined as a case where a leak is both present and detected, while a true negative corresponds to a scenario where a leak is not present and not detected. The accuracy has been computed considering the $\hat{c}$ results with respect the true label c. This metric is a common state-of-the-art metric for measuring the leak detection performances [15].

The results obtained by the baseline and the proposed methods are presented in Table I. Notably, without tuning the threshold for the new dataset, the baseline method [17] yielded a relatively low accuracy. However, by adjusting the threshold, the number of detected leaks significantly increased, reaffirming [17] as a viable solution for addressing the leak detection problem.

Leak Starting Time. To measure the performance for the starting time detection we used as metric the DD. The DD is defined as

$$\mathrm{DD} = \hat{T} - T^{\star}, \tag{6}$$

were we assume that $\hat{T} \geq T^{\star}$ as we consider the detected point before the leak happens as false positive. The DD is expressed in hours. With our pipeline we were able to achieve a DD average value of 40.21 hours.

The comparison with the baseline method [17] is provided in Table II. The DD achieved by the refined baseline is comparable to other state-of-the-art methods, despite being higher than the value reported in [17]. In contrast, our proposed method achieves a detection delay that is an order of magnitude smaller than the baselines.

Noise Robustness. To check the robustness to possible noisy data input to the algorithm, we simulated Gaussian noise at different SNR levels, which we summed to the various time series before the *Preprocessing* step. We want to highlight that the proposed method blocks were trained using only noise free signal and the noisy data were used only in test phase.

The results achieved are shown in Table III. It is possible to see that for high value of SNR we are still able to achieve good performances both in accuracy and DD. By reducing the SNR, a marked decline in performance is observed. To address this issue, potential strategies include incorporating a noise reduction stage at the initial phase of the proposed method's pipeline or training the feature extractor and one-class SNR with noisy data to enhance robustness.

TABLE III
DETECTION ACCURACY AND DELAY FOR DIFFERENT SNR LEVELS

SNR	**Accuracy**	**Detection Delay**
45 dB	0.72	81.4
40 dB	0.63	187.1
35 dB	0.63	321.5
30 dB	0.59	422.2
25 dB	0.59	454.6

TABLE IV
DETECTION ACCURACY AND DELAY FOR HANOI WDN

WDN	**Accuracy**	**Detection Delay (hours)**
Hanoi	0.77	102

Detection on Different WDNs. To evaluate the generalization capabilities of the proposed method on previously unseen WDNs, we applied it to the Hanoi and Pescara networks. These experiments utilized the LeakDB dataset [25], which provides 500 simulated scenarios, including both leak and no-leak cases, for the Hanoi network. Since the Hanoi WDN has fewer nodes than the network used during the method's development, we replicated the nodes to match the input size required by the feature extractor. The accuracy and DD values obtained are reported in Table IV.

The results are promising, particularly as the method was applied without significant adaptation to the specifics of the new network, except for adjusting the number of nodes. The method successfully detected 77% of leaks in the Hanoi network. While the DD value (102 hours) is slightly higher than those reported for state-of-the-art methods, this could be improved by optimizing the detection threshold for this specific network.

Similarly, we applied the proposed method to the Pescara WDN. The Pescara network consists of 68 junctions, 3 reservoirs, and 99 pipes. We simulated 500 scenarios, both with and without leaks, varying the leak positions and sizes for the leak scenarios. As with the Modena network, the simulations were conducted using the EPANET software, with data sampled every 30 minutes.

The results for the Pescara WDN are shown in Table V. The proposed method achieved an accuracy of 82% and a mean DD of 54 hours. These results are closer to those obtained for the Modena network, further highlighting the generalization capabilities of the proposed pipeline across different network topologies.

TABLE V
DETECTION ACCURACY AND DELAY FOR PESCARA WDN

WDN	Accuracy	Detection Delay (hours)
Pescara	0.82	54

V. CONCLUSION

In this work, we propose a novel methodology for leak analysis in a WDN. Specifically, we address two key tasks:

- Detecting the presence of a leak in a WDN.
- Determining the starting time of the leak.

The proposed methodology combines a feature extractor with a one-class SVM trained exclusively on no-leak data to identify leaks as anomalies.

To evaluate the method, we simulated various scenarios using the Modena WDN as the backbone network. Our approach demonstrates strong performance in detecting leaks. The pipeline is highly efficient, achieving an average detection delay (DD) of 40.21 hours with a detection accuracy of 92%.

To our knowledge, this is one of the first studies to comprehensively address the challenges of noisy input data and generalization capabilities in the field of water leak detection. We extensively evaluated the robustness of the proposed method across different signal-to-noise ratio (SNR) levels and assessed its generalization on two additional WDNs that were not part of the training phase. These results underscore the method's reliability and adaptability across varying network conditions.

For future work, we aim to validate the algorithm on real-world scenarios using actual pressure measurements. We also plan to integrate a leak localization component and develop advanced strategies to handle noisy input data more effectively, further enhancing the method's applicability in practical settings.

REFERENCES

[1] S. R. Mounce, J. B. Boxall, and J. Machell, "Development and Verification of an Online Artificial Intelligence System for Detection of Bursts and Other Abnormal Flows," *ASCE Journal of Water Resources Planning and Management*, pp. 309–318, 2010.

[2] J. Mashford, D. D. Silva, S. Burn, and D. Marney, "Leak detection in simulated water pipe networks using svm," *Applied Artificial Intelligence*, pp. 429–444, 2012.

[3] D. U. Leonzio, P. Bestagini, M. Marcon, G. P. Quarta, and S. Tubaro, "Water leak detection and classification using multiple sensors," in *IFIP Networking Conference*, 2024.

[4] S. Adachi, S. Takahashi, H. Kurisu, and H. Tadokoro, "Estimating Area Leakage in Water Networks Based on Hydraulic Model and Asset Information," in *Water Distribution System Analysis Conference (WDSA)*, 2014.

[5] N. Mashhadi, I. Shahrour, N. Attoue, J. El Khattabi, and A. Aljer, "Use of Machine Learning for Leak Detection and Localization in Water Distribution Systems," *Smart Cities*, pp. 1293–1315, 2021.

[6] P. Irofti, F. Stoican, and V. Puig, "Fault handling in large water networks with online dictionary learning," *Journal of Process Control*, pp. 46–57, 2020.

[7] D. U. Leonzio, P. Bestagini, M. Marco, P. Quarta, and S. Tubaro, "Water Leak Detection via Domain Adaptation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[8] T. K. Chan, C. S. Chin, and X. Zhong, "Review of Current Technologies and Proposed Intelligent Methodologies for Water Distributed Network Leakage Detection," *IEEE Access*, pp. 78 846–78 867, 2018.

[9] R. Perez, G. Sanz, V. Puig, J. Quevedo, M. A. Cuguero Escofet, F. Nejjari, J. Meseguer, G. Cembrano, J. M. Mirats Tur, and R. Sarrate, "Leak Localization in Water Networks: A Model-Based Methodology Using Pressure Sensors Applied to a Real Network in Barcelona," *IEEE Control Systems Magazine*, pp. 24–36, 2014.

[10] M. Romano, Z. Kapelan, and D. Savić, "Automated Detection of Pipe Bursts and Other Events in Water Distribution Systems," *ASCE Journal of Water Resources Planning and Management*, pp. 457–467, 2014.

[11] M. Quiñones-Grueiro, M. Milián, M. Rivero, A. Neto, and O. Llanes-Santiago, "Robust leak localization in water distribution networks using computational intelligence," *Neurocomputing*, vol. 438, pp. 195–208, 2021.

[12] J. Zhang, C. Lu, X. Li, H. Kim, and J. Wang, "A full convolutional network based on densenet for remote sensing scene classification," *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 3345–3367, 2019.

[13] C. Geelen, D. Yntema, J. Molenaar, and K. Keesman, "Monitoring support for water distribution systems based on pressure sensor data," *Water Resources Management*, vol. 33, pp. 3339–3353, 2019.

[14] L. McInnes, J. Healy, S. Astels *et al.*, "hdbscan: Hierarchical density based clustering." *J. Open Source Softw.*, vol. 2, p. 205, 2017.

[15] A. Soldevila, G. Boracchi, M. Roveri, S. Tornil-Sin, and V. Puig, "Leak detection and localization in water distribution networks by combining expert knowledge and data-driven models," *Neural Computing and Applications*, pp. 4759–4779, 2022.

[16] E. Galdiero, F. D. Paola, N. Fontana, M. Giugni, and D. A. Savić, "Decision support system for the optimal design of district metered areas," *Journal of Hydroinformatics*, pp. 49–61, 2016.

[17] D. U. Leonzio, P. Bestagini, M. Marco, P. Quarta, and S. Tubaro, "Water Leak Detection and Localization using Convolutional Autoencoders," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[18] ——, "Robust Water Leak Detection and Localization with Graph Signal Processing," in *IEEE Industrial Electronics, Control, and Instrumentation Conference (IECON)*, 2023.

[19] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," in *Arxiv*, 2020.

[20] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[21] S. Conety, T. Neelakantan, P. R. Sivakumar, and D. Páez, "Analysis of water distribution network under pressure-deficient conditions through emitter setting," *Drinking Water Engineering and Science*, vol. 12, 2019.

[22] H. Monsef, M. Naghashzadegan, A. Jamali, and R. Farmani, "Comparison of evolutionary multi objective optimization algorithms in optimum design of water distribution network," *Ain Shams Engineering Journal*, vol. 10, pp. 103–111, 2019.

[23] M. Quiñones-Grueiro, M. A. Milián, M. S. Rivero, A. J. S. Neto, and O. Llanes-Santiago, "Robust leak localization in water distribution networks using computational intelligence," *Neurocomputing*, vol. 438, pp. 195–208, 2021.

[24] L. Rossman, "Epanet 2.0 user manual," 2000.

[25] S. G. Vrachimis, M. S. Kyriakou, D. G. Eliades, and M. M. Polycarpou, "Leakdb: a benchmark dataset for leakage diagnosis in water distribution networks," in *WDSA/CCWI joint conference proceedings*, 2018.