First Results in Leak Localization in Water Distribution Networks using Graph-Based Clustering and Deep Learning ⋆

Luis Romero * **Joaquim Blesa** *,** **Vicenç Puig** *,**
Gabriela Cembrano * **Carlos Trapiello** *,**

* *Institut de Robòtica i Informàtica Industrial (CSIC-UPC). Carrer Llorens Artigas, 4-6, 08028 Barcelona (email: lromero@iri.upc.edu).*
** *Supervision, Safety and Automatic Control Research Center (CS2AC) of the Universitat Politécnica de Catalunya, Campus de Terrassa, Gaia Building, Rambla Sant Nebridi, 22, 08222 Terrassa, Barcelona.*

Abstract: This paper presents a methodology for the localization of leaks in water distribution networks (WDNs) by means of the combination of a deep learning (DL) approach and a graph-based clustering technique. A data set for all possible leak locations is generated from pressure measurements and utilized to feed an image encoding process based on the Gramian Angular Field (GAF) technique, hence producing an equivalent data set of images. The pressure measurements are generated through the WDN simulation engine EPANET. To accomplish the training stage, the network is iteratively segmented into clusters using the Graph Agglomerative Clustering (GAC) method, and a deep learning neural network (DLNN) is trained to correctly indicate the leak location at one of the created clusters. The achieved neural networks tree can be traversed through its different branches depending on each classification result, until the final cluster is reached. Consequently, leaks can be located with a success rate that grows inversely to the size of the clusters. Due to the dependency of the latter on the number of clusters, which can be settled, the presented method is adaptable to the considered network features (as e.g. dimensions, sensors placement and accuracy) and requisites (as e.g. localization area size).

Keywords: leak localization, WDN, deep learning, graph-based clustering, neural networks

1. INTRODUCTION

Leak detection and localization in water distribution networks (WDNs) is of great interest for water distribution companies because leaks in WDNs are estimated to account up to 30 % of the total amount of distributed water (Puust et al., 2010). This is one of the reasons why leak detection and localization in WDNs is a very active area of research, see Chan et al. (2018) for a recent and extensive review.

Model-based leak localization methods that use hydraulic models can provide a satisfactory efficiency, however some errors are introduced due to the presence of model errors as nodal demand uncertainty and noise in the measurements (Cugueró-Escofet et al., 2015; Blesa and Pérez, 2018). They can be taken into account in the design of machine learning based leak localization methods (Ferrandez-Gamot et al., 2015). Due to this fact and the growth of machine learning and data-driven techniques (Goodfellow et al., 2016), the challenge of achieving a consistent solution to the model-free leak localization problem starts to be seriously tackled. It may be reformulated as the design and development of a process that only utilizes the available information of the water distribution network, i.e. measured data of the network dynamical evolution, to indicate the location of a leak at a certain node, considering that the detection operation is correctly performed.

Arifin et al. (2018) presents a novel data-driven method for both leak detection and localization based on the concept of Kantorovich Distance and using mass flow rates and pressure measurements. Candelieri et al. (2014) presents a graph-based methodology that exploits spectral clustering over a graph whose nodes are the defined leak scenarios. Soldevila et al. (2019) presents a data-driven technique that estimates the pressure value at all the network nodes from the measured ones by means of the Kriging spatial interpolation method, comparing leak and leak-free scenarios to find the affected node. Parellada et al. (2019) extends the previous work, applying different classifiers to the pressure values obtained after the interpolation.

In this work, the leak localization problem is formulated as a classification task from the pattern recognition point of view. The proposed methodology is based on two main ideas:

- To make use of the power of deep learning (DL) techniques (Sengupta et al., 2019), applying them

⋆ The authors wish to thank the support received by the Spanish national project DEOCS (DPI2016-76493-C3-3-R) and by SMART Project (ref. num. EFA153/16 Interreg Cooperation Program POCTEFA 2014-2020).

to images generated by encoding the available data. Deep learning has been already utilized for leak localization tasks by Zhou et al. (2019) and Javadiha et al. (2019).
- To divide the leak localization task into a set of simple classification problems, organizing the resultant classifiers in a hierarchical way. Graph-based clustering techniques (Schaeffer, 2007) are exploited to tackle the network division process.

The main contribution of the presented approach consists of the ability to regulate the localization area (by means of the clustering approach settings: size and number of final clusters) depending on the network characteristics: topology, sensor placement and precision, etc., adapting the method performance to the considered leak isolation problem and, consequently, analysing the localization limitations for data-driven approaches. Therefore, the proposed methodology can be exploited as a complete leak localization method, as well as a decision-making tool.

2. METHODOLOGY

The design of a learning-based approach for leak localization implies that the process must be trained for its later usage. The general training procedure is represented in Figure 1.

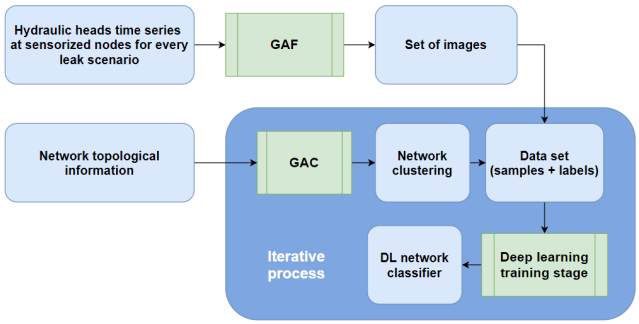


Fig. 1. Training stage - Workflow scheme

The leak localization method considers N pressure sensors installed in inner nodes of the WDN, as well as the network topology.

Pressure values and elevation at the sensorized inner nodes are collected in a vector X:

$$X = [x_1, x_2, \ldots, x_j, \ldots, x_N] \tag{1}$$

where x_j, $\forall j = 1, ..., N$, is the hydraulic head at node j, i.e., measured pressure in node j plus elevation at this node.

The hydraulic input information consists of instances of vector X defined in (1):

$$X_i^k \in \mathbb{R}^N,\ i = 1, ..., M,\ k = 1, ..., P \tag{2}$$

where i defines the time instant and k stands for the leak scenario (a leak at a concrete node of the network), and M and P are the total number of time instants and leak scenarios, respectively.

This input information is used to produce the set of samples for the training stage, composed by images obtained from an encoding process known as Gramian Angular Field or GAF, presented in Wang and Oates (2015).

To split and facilitate the complete classification problem, Graph Agglomerative Clustering (GAC) (Zhang et al. (2012) and Zhang et al. (2013)) has been applied to divide each network or subnetwork into two sets of nodes. The partition would depend on the WDN topological information provided to the algorithm.

The training process consists of an iterative procedure between clustering and deep learning neural network (DLNN) training. A classifier is obtained at each step to identify the origin of the leak between the two set of nodes generated from the clustered graph.

The achieved set of neural networks is hierarchically organized to form a classification tree. Each tested sample traverses a concrete path until the final cluster is reached, deciding at each step between two possible branches.

2.1 Data encoding

The GAF technique is utilized as the data encoding tool for the conversion of each hydraulic head vector X_i^k into its associated image. Due to its original development for time-series, its application must be adapted to handle the mentioned measurements vectors instead.

The commonly limited number of installed sensors at WDNs (Savić et al., 2009) explains this feature selection. It delimits the size of the GAF images, and hence reduces the computational cost and avoids the necessity of gathering M-dimensional vectors (considering M to be the length of the time-series data) for the localization of a single leak.

As a preprocessing, every X_i^k vector is rescaled to range the values between -1 and 1:

$$\tilde{x}_{ij}^k = \frac{\left(x_{ij}^k - max(X_i^k)\right) + \left(x_{ij}^k - min(X_i^k)\right)}{max(X_i^k) - min(X_i^k)} \tag{3}$$

The data vector $\tilde{X}_i^k$ can be encoded in polar coordinates as follows:

$$\phi_{ij}^k = arccos(\tilde{x}_{ij}^k) \tag{4}$$

$$r_{ij}^k = \frac{i}{n} \tag{5}$$

being n a constant factor to regularize the span of the polar coordinate system.

In this case, the GAF process exploits the angular perspective by considering the trigonometric sum/difference between each pair of sensor values to identify their correlation, obtaining the Gramian Angular Field as:

$$GAF_i^k = [cos(\phi_{ij_1}^k + \phi_{ij_2}^k)] = (\tilde{X}_i^k)' \cdot \tilde{X}_i^k - \left(\sqrt{I - (\tilde{X}_i^k)^2}\right)' \cdot \sqrt{I - (\tilde{X}_i^k)^2} \tag{6}$$

where j_1 and j_2 are two possible values of j and I is the unit row vector.

Then, the GAF resulting image consists of a matrix whose x-y component encodes the relation between the pressure sensors x and y. To achieve the standard format of an image, the GAF matrices are processed to locate its values in the range 0-255.

2.2 Clustering

The clustering process is utilized to split the network (or a previously computed subnetwork) into two new subnetworks, by means of the topological information of its associated graph.

The GAC algorithm has been selected to handle this operation. It is based on three main stages:

(1) The generation of a directed graph from the undirected one given as input.
(2) The utilization of the k-NN technique, described in Sutton (2012), to generate a set of initial clusters.
(3) The merger of the k-NN produced sets of nodes into larger clusters based on their affinity, until the desired number of clusters is reached.

The GAC algorithm employs the weights between nodes of the graph to decide its splitting into new subgraphs. Hence, the selection of these weights allows to regulate the criteria the algorithm employs to produce the new subnetworks.

To handle the high dependence on the value of the parameter k, the clustering process for each (sub)network is embedded in a loop that varies its value in order to obtain the maximum possible validation accuracy.

2.3 Deep learning

After a (sub)network partition, a different label is assigned to each one of the resulting clusters, and consequently, to its member nodes.

Considering a raw data set with M timesteps, N nodes and P leak scenarios, after the encoding stage, the final data set would be composed by $M \cdot P$ samples ($N \times N$ GAF matrices/images), i.e.

$$GAF_i^k \in \mathbb{R}^{N \times N},\ i = 1, ..., M,\ k = 1, ..., P \qquad (7)$$

These samples, together with their corresponding labels, must be divided into training, validation and testing sets.

The structure of the generated DLNN is displayed in Figure 2.

The input image layer size is $N \times N$ (the depicted input matrix G is an instance of GAF_i^k for a concrete time step and leak scenario). It is followed by three instances of a set of layers composed by:

- A convolutional layer (Murphy, 2016), that applies sliding filters to the input image (regarding the small number of sensors and hence the low value of N, the sizes of the sliding filters of the different convolutional layers have a low upper limit).

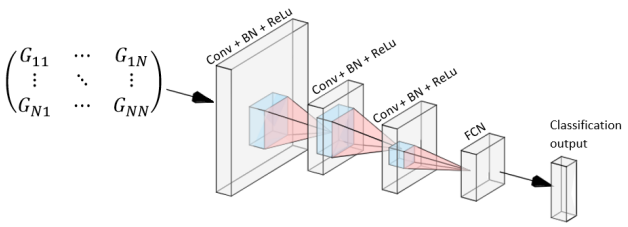


Fig. 2. DL network structure

- A batch normalization layer (Ioffe and C.Szegedy, 2015) to speed up the training phase, reduce the dependence of the network to the weight initialization, as well as reduce overfitting.
- A ReLu layer (Agarap, 2018) as non-linear output.

A fully connected (FC) layer with an output layer of size 2 is employed to achieve the binary classification (a softmax layer (Bishop, 2006) is included to be the output unit activation function after this FC layer).

The usage of more complex sets of layers does not bring about an improvement of the classification results, and hence a straightforward solution was adopted to speed up the training stage. In the same way, the number of instances of the presented set of layers is derived from the experiments performance. The inclusion of a smaller number of them yields to poorer results, whereas the usage of extra instances does not enhance the classification.

3. CASE STUDY

To validate the proposed method, a real network has been utilized as example. It is a District Metering Area (DMA) that presents 121 nodes and 125 pipes. It is graphically represented in Figure 3.

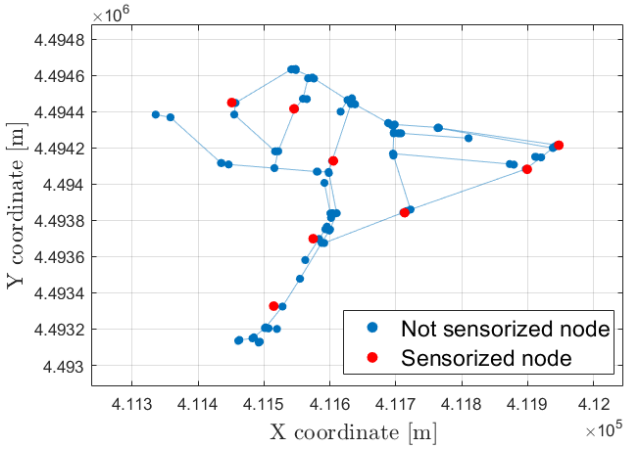


Fig. 3. Case study WDN (with sensorized nodes)

3.1 Data encoding in the case study

The GAF method is applied to convert the eight head values at the sensorized nodes into an 8×8 image.

Figures 4 and 5 respectively show the location of various nodes of the network and the GAF images that are associated to a leak event at each one of them.

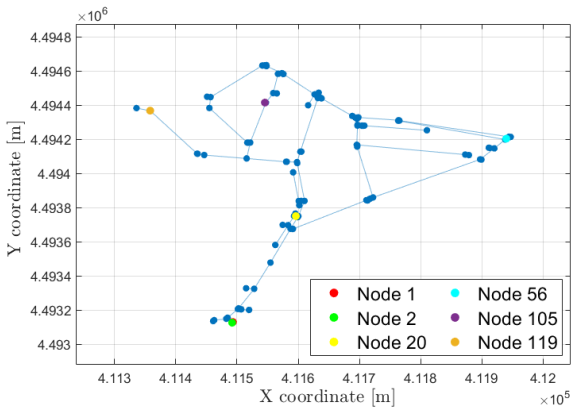
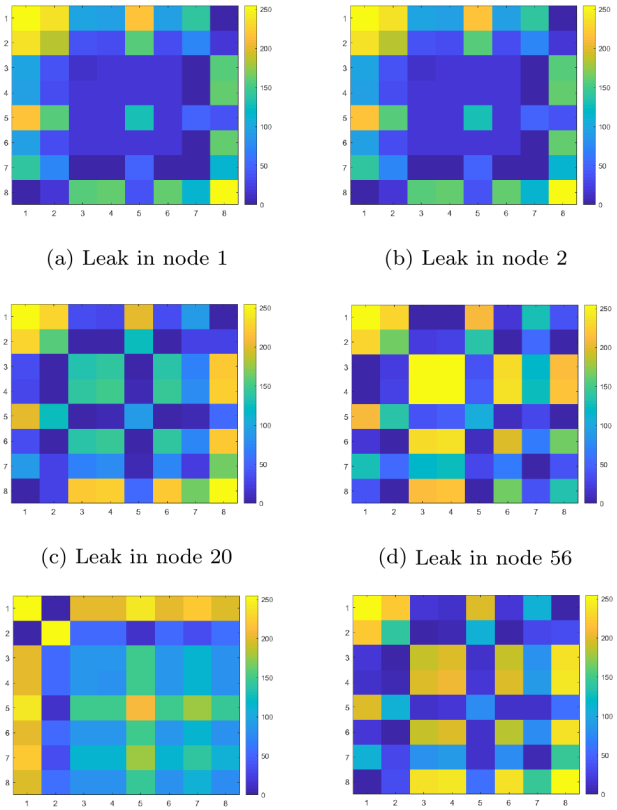


Fig. 4. Case study WDN (with GAF images nodes)



(a) Leak in node 1 (b) Leak in node 2

(c) Leak in node 20 (d) Leak in node 56

(e) Leak in node 105 (f) Leak in node 119

Fig. 5. GAF images associated to different leaks

On the one hand, most images corresponding to leaks at close nodes, like 1 and 2 (Figures 5a and 5b respectively), share a high degree of resemblance. This leads to a rather difficult distinction between them.

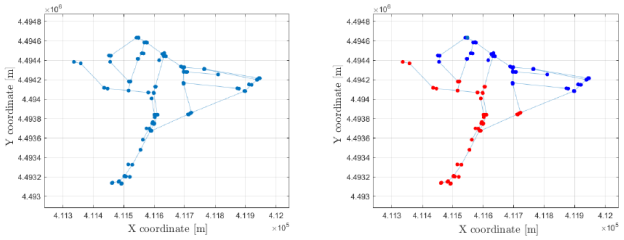
On the other hand, for the case of most sufficiently distant nodes, differences between the images become evident, facilitating the leak location task.

Therefore, the indistinguishability among nodes tends to become higher once a certain level of proximity is reached. This fact supports the decision of exploiting a clustering approach to decrease the classification difficulty.

A group of nodes whose classification becomes a hard problem would be considered as a unique cluster, whereas sets of nodes with intra-distinguishability can still be clustered. In this way, the similarity among images determines the clustering limit, depending the former on the network and sensor characteristics.

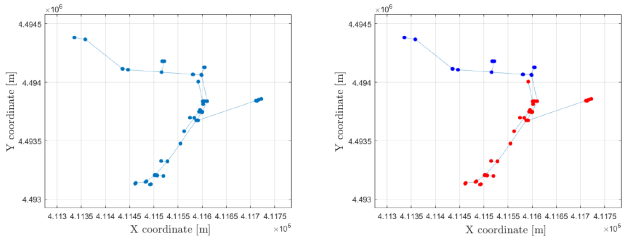
3.2 Clustering in the case study

The clustering task is tackled by the Graph Agglomerative Clustering algorithm.



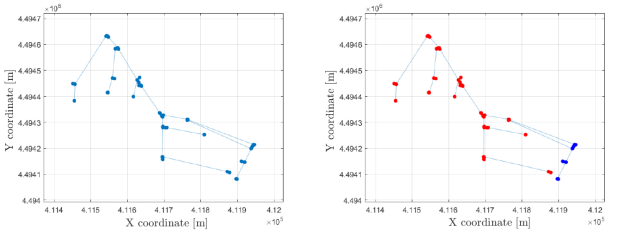
(a) Complete network (b) Clustered network

Fig. 6. Clustering - Complete network



(a) First subnetwork (b) Clustered first subnetwork

Fig. 7. Clustering - First subnet from complete network



(a) Second subnetwork (b) Clustered second subnetwork

Fig. 8. Clustering - Second subnet from complete network

Figure 6 shows the clustering process of the original network. Concretely, Figure 6a depicts the complete case study network before the clustering, and Figure 6b presents the two generated clusters. Figures 7 and 8 display the analogue clustering process for the two previously produced subnetworks.

Therefore, this process is carried out iteratively and recursively, as it is applied to the results of previous instances of the method.

The value of the k parameter of the GAC method for an actual usage of the localization algorithm, as aforementioned, is automatically settled depending on the achieved localization accuracy.

There are various decisions about the clustering process:

- The pipe distance between nodes of the WDN is fed to the clustering algorithm as the metric that represents the relation between each pair of nodes, regulating the division of the nodes into sets. Additional topological information can be provided to the technique in order to modify the clustering behaviour.
- The clustering limit needs to be decided considering the level of indistinguishability among the nodes of the analysed network, i.e., the number and size of clusters must depend on the network features and how they affect the localization accuracy.

3.3 Deep learning in the case study

The MATLAB® Deep Learning Toolbox™ has been utilized to deploy the training, validation and testing stages. Concretely, each DLNN is trained using the following learning settings (see MathWorks (2019) for more information about the different parameters):

Table 1. DL training settings

Parameter	*Value*
Solver	SGDM
Initial learning rate	0.01
Learning rate schedule	piecewise
Learning rate dropping period	8
Learning rate dropping factor	0.9
Maximum number of epochs	150
Convolutional filter size	$\left[\lfloor N/2 \rfloor \ \lfloor N/3 \rfloor \ \lfloor N/4 \rfloor\right]$
Number of filters	[8 16 32]

The last two elements of the table are vectors because each component represent the value of the parameter for one of the three set of layers forming the DLNN structure.

The data sets are divided into a 75% for training, 15% for validation and 10% for testing.

The training and clustering stages are integrated into an iterative process. When a (sub)network is clustered into two new subnetworks, a DLNN is trained to distinguish the leak location between these new subnetworks.

To guarantee a sufficiently good performance of the classifiers (and to settle a proper k parameter, as aforementioned), the clustering and learning processes of a certain (sub)network are repeated until a minimum localization accuracy is reached at the validation phase.

3.4 Results

To test the presented methodology, the necessary hydraulic results are obtained using the WDN modelling and simulation software EPANET 2 (L. A. Rossman, 2000). One simulation is carried out for each possible scenario, i.e., producing a leak at each one of the network nodes. Each simulation comprises 96 hours, with a new measurement every 2 minutes. However, to reduce the training duration, only a sample per hour is utilized. The leak size has been fixed to 1 l/s.

From each complete simulation, only 24 hours are employed for the learning stage (this includes the training, validation and testing phases). Additional 24 hours are utilized for further testing of the procedure.

The obtained performance results for different clustering approaches are presented in Table 2 by means of the following parameters:

- The number of final clusters (first column).
- The leak localization accuracy (second column), i.e., the percentage of leaks that are correctly located at the corresponding node or cluster.
- The mean of the Average Topological Distance or ATD (third column) from the leaking node to the set of nodes belonging to the reached cluster through the obtained path. For further explanations of this metric, see Javadiha et al. (2019).
- The mean value of the cluster size (fourth column), expressed as the number of member nodes.
- The mean value of the cluster size (fifth column), expressed as the radius of the minimum circle enclosing all the member nodes.

Table 2. Performance results

N_c	$Acc(\%)$	$\overline{ATD}$	$\overline{C_{size}}\ (n^o nodes)$	$\overline{C_{size}}\ (m)$
8	100	0	15.00	146.97
16	99.17	0.04	7.50	63.78
30	91.22	0.32	4.00	34.26
57	81.88	0.67	2.11	6.66
87	73.72	0.79	1.38	2.30

The results show that the classification performance gets deteriorated as the number of clusters increases. The accuracy is the most affected metric, whereas the ATD do not greatly augment. The latter implies that, in average, if the method fails, the actual leaking node is located at an acceptably reduced number of nodes.

In order to delve into the method behaviour, uniform random noise is added to the nominal measurements. The associated results for noise ranges of $\pm 0.01\ m$ and $\pm 0.1\ m$ are presented in Table 3 and Table 4 respectively.

Table 3. Performance results - $\pm 0.01\ m$

N_c	$Acc(\%)$	$\overline{ATD}$	$\overline{C_{size}}\ (n^o nodes)$	$\overline{C_{size}}\ (m)$
8	93.96	0.38	15.00	92.16
16	85.31	0.68	7.50	49.63
30	77.22	1.25	4.00	24.72
46	70.56	1.55	2.61	14.21
80	44.79	2.56	1.50	3.06

Table 4. Performance results - $\pm 0.1\ m$

N_c	$Acc(\%)$	$\overline{ATD}$	$\overline{C_{size}}\ (n^o nodes)$	$\overline{C_{size}}\ (m)$
8	73.30	2.94	15.00	82.07
16	57.19	4.52	7.50	55.65
30	48.13	5.36	4.00	24.62
46	38.02	6.44	2.61	13.07
75	24.72	6.65	1.60	3.49

A noise reduction approach is applied to decrease the uncertainty effects. The complete data set, with a sample every 2 minutes, is averaged every 30 elements. Hence, each final value represents the mean over 1 hour.

The degeneration of the performance is caused by the slight differences between data vectors of different leak scenarios, which are the learning objective of the deep learning stage. Therefore, when the sensor noise is as wide as those differences, the performance is critically worsen. Analysing these differences for the complete data

set (before averaging), only a 25% of them are superior than an uncertainty tolerance of ± 0.1 m, as well as a 86.7% in the case of a noise range of ± 0.01 m. These two facts completely explain the cause of the deterioration of the leak localization performance. The areas comprising indistinguishable nodes become larger due to the noise level overlapping the differences among nodes in their behaviour in the presence of a leak.

4. CONCLUSIONS

The first results presented in this work indicate the procedure potential benefits, allowing to regulate the leak localization area depending on the network characteristics. This leads to the design of an ad hoc solution for the leak localization problem at each concrete WDN, regarding the data availability, the network topology and the sensors location and precision. Regarding the latter, a trade-off between the classification performance and the clustering suitability must be considered in the presence of sensor noise ranges that are too large in comparison with the differences among leak scenarios. Further steps can be taken to extend and enhance the methodology.

- The clustering strategy divides the (sub)network into two clusters iteratively, generating a high number of final neural networks. It would be interesting to analyse the effect of increasing the number of resulting clusters after each division.
- The clustering process might be enhanced providing additional information, like the sensor locations.
- The generation of a single classifier with as many outputs as the desired number of clusters can be explored. The insight about the network limitations, gained from the hierarchical approach presented in this work, would accelerate the design process.
- An exhaustive assessment of the methodology suitability and limitations should be addressed: exploring a higher variability of scenarios, delving into the influence of the experimental settings as well as the noise levels and comparing the presented approach with other state-of-the-art techniques.

REFERENCES

Agarap, A. (2018). Deep learning using rectified linear units (relu).

Arifin, B.M.S., Li, Z., Shaha, S.L., Meyer, G.A., and Colin, A. (2018). A novel data-driven leak detection and localization algorithm using the Kantorovich distance. *Computers and Chemical Engineering*, 108, 300–313.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer, New York, NY.

Blesa, J. and Pérez, R. (2018). Modelling uncertainty for leak localization in water networks. *IFAC-PapersOnLine*, 51(24), 730 – 735.

Candelieri, A., Conti, D., and Archetti, F. (2014). A graph based analysis of leak localization in urban water networks. *Procedia Engineering*, 70, 228–237.

Chan, T.K., Chin, C.S., and Zhong, X. (2018). Review of current technologies and proposed intelligent methodologies for water distributed network leakage detection. *IEEE Access*, 6, 78846–78867.

Cugueró-Escofet, P., Blesa, J., Pérez, R., Cugueró-Escofet, M.A., and Sanz, G. (2015). Assessment of a leak localization algorithm in water networks under demand uncertainty. *IFAC-PapersOnLine*, 48(21), 226 – 231.

Ferrandez-Gamot, L., Busson, P., Blesa, J., Tornil-Sin, S., Puig, V., Duviella, E., and Soldevila, A. (2015). Leak localization in water distribution networks using pressure residuals and classifiers. *IFAC-PapersOnLine*, 48(21), 220 – 225.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*, chapter 1, 11–28. MIT Press.

Ioffe, S. and C.Szegedy (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML'15 Proceedings of the 32nd International Conference on Machine Learning*, 37.

Javadiha, M., Blesa, J., Soldevila, A., and Puig, V. (2019). Leak localization in water distribution networks using deep learning. *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 1426–1431.

L. A. Rossman (2000). EPANET 2 User's Manual. *U.S. Environmental Protection Agency, Washington, D.C., EPA/600/R-00/057*.

MathWorks (2019). MATLAB® Deep Learning Toolbox - Training options. `https://es.mathworks.com/help/deeplearning/ref/trainingoptions.html`.

Murphy, J. (2016). An overview of convolutional neural network architectures for deep learning.

Parellada, B., Sun, C., Puig, V., and Cembrano, G. (2019). Leak localization in water distribution networks using pressure and a data-driven classifier approach. Technical Report IRI-TR-19-04, Institut de Robòtica i Informàtica Industrial - CSIC-UPC.

Puust, R., Kapelan, Z., Savić, D.A., and Koppel, T. (2010). A review of methods for leakage management in pipe networks. *Urban Water Journal*, 7(1), 25–45.

Savić, D., Kapelan, Z., and Jonkergouw, P. (2009). Quo vadis water distribution model calibration? *Urban Water Journal*, 6, 3–22.

Schaeffer, S.E. (2007). Graph clustering. *Computer Science Review*, 1, 27–64.

Sengupta, S., Basak, S., Saikia, P., Paul, S., Tsalavoutis, V., Atiah, F., Ravi, V., and Peters, A. (2019). A review of deep learning with special emphasis on architectures, applications and recent trends.

Soldevila, A., Blesa, J., Fernández-Canti, R.M., Tornil-Sin, S., and Puig, V. (2019). Data-driven approach for leak localization in water distribution networks using pressure sensors and spatial interpolation. *Water*, 11(7), 1500.

Sutton, O. (2012). Introduction to k nearest neighbour classification and condensed nearest neighbour data reduction.

Wang, Z. and Oates, T. (2015). Imaging time-series to improve classification and imputation.

Zhang, W., Wang, X., Zhao, D., and Tang, X. (2012). Graph degree linkage: Agglomerative clustering on a directed graph. *ECCV*.

Zhang, W., Zhao, D., and Wang, X. (2013). Agglomerative clustering via maximum incremental path integral. *Pattern Recognition*, 46(11), 3056–3065.

Zhou, X., Tang, Z., Xu, W., Meng, F., Chu, X., Xin, K., and Fu, G. (2019). Deep learning identifies accurate burst locations in water distribution networks. *Water Research*, 166.