

# Learning Dictionaries from Physical-Based Interpolation for Water Network Leak Localization

Paul Irofti, *Member, IEEE*, Luis Romero-Ben, Florin Stoican, *Member, IEEE*, and Vicenç Puig

**Abstract**—This article presents a leak localization methodology based on state estimation and learning. The first is handled by an interpolation scheme, whereas dictionary learning is considered for the second stage. The novel proposed interpolation technique exploits the physics of the interconnections between hydraulic heads of neighboring nodes in water distribution networks. Additionally, residuals are directly interpolated instead of hydraulic head values. The results of applying the proposed method to a well-known case study (Modena) demonstrated the improvements of the new interpolation method with respect to a state-of-the-art approach, both in terms of interpolation error (considering state and residual estimation) and posterior localization.

**Index Terms**—leak localization, water distribution network, dictionary learning, state estimation, interpolation

## I. INTRODUCTION

**T**HE efficient management of water resources plays a crucial role in modern society, as clean water is vital for human life. Despite this, around four billion people suffer from severe water scarcity during at least one month per year [1]. This will only get worse in the near future, considering the continuous growth in the world-wide population, the climate change and the massification of urban areas [2].

Consequently, water utilities are greatly interested in developing methodologies for the efficient distribution of water. One of their main interests is the monitoring of leaks over water distribution networks (WDNs), which are estimated to exceed 126 billion cubic meters per year worldwide [3]. This deeply impacts society, not only due to the economical and operational cost, but because of the environmental [4] and sanitary [5] repercussions. The research on this field has evolved from initial works [6], [7], rapidly growing and leading to a wide range of methods (see [8] for a recent review about leak detection and localization methods).

Model-based schemes use calibrated hydraulic models (in terms of network properties and nodal consumption) to compare simulated hydraulic data with measurements from the real

network. This approach goes back to the first works on the subject, when analysis of pressure sensitivity to the existence of leaks was studied [7]. The concept of sensitivity was further exploited in well-known articles, by means of a fault-sensitivity-matrix (FSM) that stores the effect of every possible leak on each junction [9], [10]. Additional techniques like the solution of the inverse problem, which consists of calibrating network parameters and demands from pressure/flow data, have also been widely used to detect/locate leaks [11], [12]. Bayesian-based approaches [13] or fuzzy logic [14] have also been explored. Their effectiveness is limited due to the diversity and complexity of existent networks [15], the difficult selection/calibration of the required hydraulic models [16] and the existence of modeling errors [17].

Recently, data-driven methods have started to be considered in order to circumvent the necessity of a hydraulic model, with the aim of avoiding the model-related drawbacks. Most of them rely on the exploitation of hydraulic data from deployed sensors and the network's topological features. For instance, interpolation-based techniques [18], [19] estimate the values of unknown hydraulic variables from the set of measured ones; and graph-based methods [20], [21] exploit graph-related techniques (e.g. clustering) or properties to help solving the localization task. Their performance is satisfactory, particularly when the goal is to identify a leak search area over the network, even more when considering their independence of hydraulic models. Nevertheless, their accuracy tends to be lower in comparison to model-based schemes when their aim is to find the exact leak location. Moreover, the critical dependency on hydraulic measurements may lead to the necessity of a large set of sensors, increasing the operational cost.

The existing drawbacks of model-based and data-driven approaches, together with the development of data analysis and machine learning algorithms, led to the appearance of mixed model-based/data-driven techniques. They use those algorithms to reduce the dependence on a hydraulic model, while maintaining the node-level accuracy. Different algorithms, such as artificial neural networks (ANNs) [22], support vector machine (SVM) [23], deep learning [24] and dictionary learning (DL) [25]; have been studied.

This paper develops a mixed model-based/data-driven methodology based on a novel hydraulic state interpolation technique, related to the one presented in [19], and dictionary learning [26]. This work continues the research presented in [27] introducing a new physical-based interpolation method and significantly improving on the simulation and related analysis. A set of physical-based weights are derived from the WDN properties, namely pipe connectivity, lengths, diameters

Paul Irofti is with the Research Center for Logic, Optimization and Security (LOS), Department of Computer Science, Faculty of Mathematics and Computer Science, University of Bucharest, 010014 Romania, and with the Institute for Logic and Data Science, Popa Tatu 18, Bucharest, 010805 Romania, e-mail: paul@irofti.net

Luis Romero-Ben and Vicenç Puig are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028, Barcelona, Spain, e-mail: {luis.romero.ben,vicenc.puig}@upc.edu

Vicenç Puig is also with the Supervision, Safety and Automatic Control Research Center (CS2AC) of the Universitat Politècnica de Catalunya, Campus de Terrassa, Gaia Building, Rambla Sant Nebridi 22, Terrassa, 08222, Barcelona, Spain

Florin Stoican is with the Department of Automation Control and Systems Engineering, Politehnica University of Bucharest, 060042 Romania, e-mail: florin.stoican@upb.ro.

and roughness. These weights are used to pose a quadratic programming problem, which retrieves the pressure residuals (difference of pressure between leak and leak-free scenarios) for the complete network by means of the known values from sensorized nodes. Then, a group of these residuals are selected to be the samples provided to the learning stage, considering both the actual deployed sensors and a list of virtual sensors (VSs). The specific leak event linked to each selected residual is utilized to generate labels for the supervised training. Thus, the goal of combining interpolation and learning is the increase on the information provided to the DL approach (in comparison to merely supplying measurements), and the achievement of a satisfactory node-level localization accuracy that might be difficult for purely data-driven techniques while minimizing the usage of the hydraulic model (it is only necessary to obtain a set of leak samples). In this way, the strengths of each interpolation and learning/classification are boosted, while their weaknesses are diminished.

#### Contributions:

The novel methodology presented in [27] has been updated and improved in this work, leading to a set of advantages related to performance and implementation:

- The methodology presented in [27] combined an existing interpolation method [19] with dictionary learning [26]. This interpolation scheme is based on a graph weighting approach that only considers structural features (pipe lengths). This may be desirable to reduce the scheme's information needs, but degrades the resulting interpolation performance. In this paper, a new physical-based weight generation process is introduced, whereas the quadratic programming problem posed in previous articles is reformulated to fit the derivations of the new physical-based formulation. In this way, both interpolation and localization errors are reduced in comparison to the results provided by the original interpolation.
- Preliminary results in [27] encouraged the use of VSs data obtained via interpolation in the dictionary learning process. The main issue we faced revolved around the errors introduced by interpolated data, resulting in dictionaries modeling water networks that are different from the actual physical network under analysis (i.e. the process was not robust to interpolation noise). Tackling this artifact involved choosing only a few virtual sensors through human in the loop or grid-search techniques, because increasing the number of VSs would cause a loss of FDI performance. In this paper, the introduced theoretical approach producing analytical weights (used in the physical-based interpolation) allows any or all the non-sensorized nodes to be used as VSs in the dictionary learning process. Thus, the VSs are selected using purely data-driven sensor placement, unlike in [27]. The method presented in [28] is adapted to seek the set of VSs that, added to the real sensors, minimizes a pre-defined distance-based metric, computed considering the distance from all the nodes to the sensors.
- Furthermore, the interpolation-learning scheme proves its capability of balancing out the main downside of a state-of-the-art data-driven methods (including the approach in

[27]), namely the unreliable localization at node-level.

**Notations.** Vectors and matrices will be denoted in bold with lowercase and, respectively, uppercase letters. Scalars will be denoted with plain lowercase letters. For matrix  $M$  we denote  $m_i$  as its  $i^{\text{th}}$  column and  $m_{ij}$  as the entry of coordinates  $i-j$ . We denote approximations with a hat:  $\hat{M}$  is the approximation of  $M$ . The notation  $|\cdot|$  indicates cardinality when applied to sets, and absolute value otherwise.

**Outline.** The remainder of the paper is structured as follows. Section II presents the preliminary background regarding the considered interpolation techniques. Section III contains the theoretical contribution consisting of the derivation of physically-based weights, and their integration in the interpolation process together with the leak isolation mechanism via dictionary learning. The methodology from Section IV presents the integration of the interpolation and isolation process together with the most important implementation aspects. Then, Section V introduces the considered case study, presenting the specific implementation details of the method for the designed experiments. Obtained results for these experiments are subsequently exposed for both the interpolation and the localization performance. Finally, Section VI draws several conclusions about the proposed methodology, regarding its contributions, its applicability and future work-lines.

## II. PRELIMINARIES

Arguably, all leak management methodologies crucially depend on the accuracy and quantity of hydraulic information, which can be obtained from hydraulic models and/or a sufficiently large and well-distributed set of sensors. However, these can be limited by the operational and economical costs of creating/calibrating models and installing/maintaining sensors. A compromise solution lies in interpolation algorithms, which estimate the WDN hydraulic state exploiting the available WDN structure and hydraulic data collected from a reduced set of sensors [19], [29]. Let us underscore key aspects regarding these two sources of information of the network:

- The WDN structure can be modeled through a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V}$  denotes the set of nodes (reservoirs and junctions); and  $\mathcal{E}$  is the set of edges (pipes). An arbitrary node is denoted as  $v_i \in \mathcal{V}$ , while an arbitrary edge is referred to as  $e_k = (v_i, v_j) \in \mathcal{E}$ , representing the link between nodes  $v_i$  and  $v_j$ , considering the first as the source and the second as the sink. The network is composed by  $n = |\mathcal{V}|$  nodes and  $m = |\mathcal{E}|$  edges. In practice, network connectivity and pipe lengths are normally available to water utilities (considering tools like geographical information systems (GIS) [30]), while pipe diameters and roughness can be approximately estimated.
- Normally, interpolation-based schemes consider nodal pressure or, commonly, the hydraulic head (nodal pressure plus elevation), as representative of the network state. This is justified by the fact that leaks manifest as pressure drops, and pressure sensors are cheaper and easier to install than other metering devices, such as flow sensors. Thus, a set of  $n_c$  pressure sensors must be installed across the network, leading to a set of measured hydraulic heads.

Note that the heads of connected nodes are related by the flow conveyed from source to sink through a non-linear equation such as the Hazen-Williams formula [31]. This relation implies that the hydraulic head of the source will always be higher than the one of the sink, if no active elements such as pumps are installed.

All these aspects were considered to develop the Graph-based State Interpolation (GSI) technique from [19]. The core idea was to approximate the aforementioned non-linear node head dependency by a linear relation,

$$\hat{\psi}_i = \frac{1}{\phi_i} \omega_i \hat{\psi}, \quad (1)$$

where  $\hat{\psi} \in \mathbb{R}^n$  stands for the complete state vector, which approximates the actual hydraulic head vector through a mix of estimated and known values.  $\omega_i$  is the  $i$ -th row of the weighted adjacency matrix  $\Omega \in \mathbb{R}^{n \times n}$ , which encodes the weights of the connection between nodes. Specifically, GSI (as defined in [19]) gives higher weights to closer neighbors by taking them as the inverse of the actual pipe lengths

$$\omega_{ij} = \begin{cases} \frac{1}{p_k}, & \text{if } e_k = (v_i, v_j) \in \mathcal{E}, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where  $p \in \mathbb{R}^m$  is the pipe lengths vector. Moreover,  $\phi_i$  stands for the  $i$ -th element of the diagonal of the degree matrix  $\Phi \in \mathbb{R}^{n \times n}$ , directly obtained from  $\Omega$  by taking  $\phi_i = \sum_{j=1}^n \omega_{ij}$ .

The flow directionality in WDNs is imposed by the non-linear relation between hydraulic heads of adjacent nodes, with flow traversing from source to sink. This directionality can be encoded by means of the edge-node incidence matrix  $\Lambda \in \mathbb{R}^{m \times n}$ , whose entries are defined as follows

$$\lambda_{kj} = \begin{cases} 1, & \text{if } e_k = (v_i, v_j) \in \mathcal{E}, \\ -1, & \text{if } e_k = (v_j, v_i) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Nevertheless, water utilities do not usually have access to a sufficient amount of flow data to obtain an accurate hydraulic-based incidence matrix. Thus, GSI considers a structural approach to construct an approximated incidence matrix  $\hat{\Lambda}$ , as detailed in Algorithm 1.

Note that  $\mathcal{V}_R, \mathcal{V}_N$  stand for the sets of reservoirs and inner nodes respectively<sup>1</sup>, and  $shortest\_path(\cdot, \cdot, \cdot)$  generates the shortest path in a graph between two nodes [32], encoded as the output sequence/path  $\mathfrak{s}^{path}$  (step 5). Additionally,  $U \in \mathbb{R}^{n \times n}$  stores the number of times that a pipe has been considered in each  $\mathfrak{s}^{path}$  (step 7). Thus, the most used direction for each pipe is considered and encoded into the generated incidence matrix  $\hat{\Lambda}$  (steps 13 and 15). In summary, the heuristic under consideration poses that if a greater number of paths have traversed a pipe in a specific direction, it is reasonable to assume that the flow within the pipe will predominantly follow that same direction.

Both weight selection and directionality requirements are integrated in an optimization problem whose solution is the

<sup>1</sup>The algorithm exploits the fact that water must flow from the reservoirs to the rest of the water-demanding nodes.

---

### Algorithm 1 Approximated incidence matrix generation

---

**Require:**  $\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{V}_R \subseteq \mathcal{V}$

- 1: Initialize  $U = \mathbf{0}_{(n \times n)}, \hat{\Lambda} = \mathbf{0}_{(m \times n)}$
- 2: Compute  $\mathcal{V}_N = \mathcal{V} \setminus \mathcal{V}_R$
- 3: **for all**  $v_R \in \mathcal{V}_R$  **do**
- 4:   **for all**  $v_N \in \mathcal{V}_N$  **do**
- 5:      $\mathfrak{s}^{path} = shortest\_path(\mathcal{G}, v_R, v_N)$
- 6:     **for**  $i = 1, \dots, |\mathfrak{s}^{path}| - 1$  **do**
- 7:        $U(\mathfrak{s}_i^{path}, \mathfrak{s}_{i+1}^{path}) = U(\mathfrak{s}_i^{path}, \mathfrak{s}_{i+1}^{path}) + 1$
- 8:     **end for**
- 9:   **end for**
- 10: **end for**
- 11: **for all**  $e_k = (v_i, v_j) \in \mathcal{E}$  **do**
- 12:   **if**  $U(v_i, v_j) > U(v_j, v_i)$  **then**
- 13:      $\hat{\lambda}_{ki} = -1; \hat{\lambda}_{kj} = 1$
- 14:   **else**
- 15:      $\hat{\lambda}_{ki} = 1; \hat{\lambda}_{kj} = -1$
- 16:   **end if**
- 17: **end for**
- 18: **return**  $\hat{\Lambda}$

---

interpolated graph state (approximated hydraulic heads), [19]:

$$\min_{\hat{\psi}} \frac{1}{2} [\hat{\psi}^T \mathbf{L} \Phi^{-2} \mathbf{L} \hat{\psi} + \mu \gamma^2], \quad (4a)$$

$$\text{s.t. } \hat{\Lambda} \hat{\psi} \leq \mathbf{1}_n \cdot \gamma, \quad (4b)$$

$$\gamma > 0, \quad (4c)$$

$$\mathbf{Z} \hat{\psi} = \hat{\psi}_\zeta. \quad (4d)$$

The first term of objective (4a) comes from enumerating a relaxation of (1) for all nodes in  $\mathcal{V}$ . That is, we take the quadratic error of the left and right-sides of (1):

$$\sum_{i=1}^n \left[ \hat{\psi}_i - \frac{1}{\phi_i} \omega_i \hat{\psi} \right]^2 = \hat{\psi}^T \mathbf{L} \Phi^{-2} \mathbf{L} \hat{\psi}, \quad (5)$$

which becomes the term in (4a) when taking  $\mathbf{L} = \Phi - \Omega$ , i.e., the Laplacian of  $\mathcal{G}$ ; and after standard linear algebra manipulations.

The second term of (4a) and the constraint in (4b) are directly related to the flow directionality. In vector  $\Lambda \psi$ , the  $k$ -th term contains  $\lambda_{ki} \psi_i + \lambda_{kj} \psi_j$ , which is negative as per the construction in (3), and hence we expect  $\Lambda \psi \leq 0$ . However, we do not have access to the actual flow directionalities (we have instead  $\hat{\Lambda} \hat{\psi}$ ), so we relax the initial inequality to the form from (4b) via the positive slack term  $\gamma$  (4c), which is then minimized through (4a). The relative importance of interpolation error and flow directionality consistency is tuned by parameter  $\mu$ .

Finally, the available hydraulic information  $\hat{\psi}_\zeta$  is provided through the equality constraint in (4d), which restricts the state of the sensorized nodes to be equal to the actual hydraulic heads at the corresponding junctions. Matrix  $\mathbf{Z} \in \mathbb{R}^{n_\zeta \times n}$  is defined so that  $z_{oj} = 1$  only if the  $o$ -th sensor is located in node  $v_j$ , and 0 otherwise.

**Remark 1.** *The interpolation of vertex values over a graph*

following smoothing strategies was studied in the past [33]. These techniques solve a problem that intends to minimize an analogue expression to  $\frac{1}{2}[\hat{\psi}^T \mathbf{L} \hat{\psi}]$ . In this manner, the harmonic property of functions in graphs is explicitly pursued, i.e., the states of adjacent nodes are aimed to be as similar as possible (GSI does not impose this behavior, giving room to a higher degree of freedom to fulfill the added constraints).  $\blacklozenge$

**Remark 2.** Since GSI solves the quadratic problem (4), it has polynomial complexity regarding problem dimension (e.g., solving  $N$  problems with the interior point method gives  $O(n^{3.5}N)$ ).  $\blacklozenge$

### III. PROPOSED APPROACH

#### A. Physical-based interpolation

The weighting selection in GSI, described by (2), arises from the aim of minimizing the information requirements regarding the network characteristics. To this end, the network connectivity and pipe lengths are its unique inputs related to the WDN topology. However, this degrades the interpolation performance due to the approximations introduced by the weighted linear expression that substitutes the non-linear equation relating the hydraulic heads of adjacent nodes.

Thus, we aim to derive a new weighting approach to improve the interpolation, which better approximates the non-linear nature of the steady-state head-flow equations. To this end, let us recall the empirical Hazen-Williams formula<sup>2</sup>

$$\psi_i - \psi_j = \frac{10.67 \cdot p_{(i,j)}}{r_{(i,j)}^{1.852} \cdot \delta_{(i,j)}^{4.87}} \cdot f_{(i,j)} \cdot |f_{(i,j)}|^{0.852}, \quad (6)$$

where  $p_{(i,j)} = p_k$  is the length in  $[m]$ ,  $\delta_{(i,j)} = \delta_k$  is the diameter in  $[m]$ ,  $r_{(i,j)} = r_k$  is the adimensional pipe roughness coefficient and the flow  $f_{(i,j)} = f_k, \forall e_k \in \mathcal{E}$  is measured in  $[m^3/s]$ . Noting that  $\text{sign}(f_{(i,j)}) = \text{sign}(\psi_i - \psi_j)$  and that  $x = |x| \cdot \text{sign}(x)$ , we reformulate (6) into

$$f_{(i,j)} = \sigma_{(i,j)}^{0.54} \text{sign}(\psi_i - \psi_j) |\psi_i - \psi_j|^{0.54}, \quad (7)$$

where  $\sigma_{(i,j)} = \sigma_k = (r_{(i,j)}^{1.852} \cdot \delta_{(i,j)}^{4.87}) / (10.67 \cdot p_{(i,j)})$  denoting the pipe conductivity and considering that  $0.54 \approx 1/1.852$ .

In order to facilitate the manipulation of  $\text{sign}(\psi_i - \psi_j)$  through the rest of the paper, we convert the previously presented edge-node incidence matrix  $\mathbf{\Lambda}$  into node-node incidence matrix  $\mathbf{B} \in \mathbb{R}^{n \times n}$ , which is defined as

$$b_{ij} = \begin{cases} 1, & \psi_i \geq \psi_j; \\ 0, & \text{if nodes } i \text{ and } j \text{ are not connected;} \\ -1, & \psi_i < \psi_j. \end{cases} \quad (8)$$

**Remark 3.** To clarify the relationship between  $\mathbf{\Lambda}$  and  $\mathbf{B}$ , define  $\mathbf{\Lambda}_{out} = \frac{\mathbf{\Lambda} + |\mathbf{\Lambda}|}{2}$  and  $\mathbf{\Lambda}_{in} = -\frac{\mathbf{\Lambda} - |\mathbf{\Lambda}|}{2}$  as representing the inner and outer edges such that  $\mathbf{\Lambda} = \mathbf{\Lambda}_{out} - \mathbf{\Lambda}_{in}$ . Let  $\mathbf{A} = \mathbf{\Lambda}_{out}^T \mathbf{\Lambda}_{in}$  which gives  $\mathbf{B} = \mathbf{A}^T - \mathbf{A}$ . With a few linear algebra

<sup>2</sup>We redefine the pipe-related variables as follows: considering that  $e_k = (v_i, v_j)$ , the  $k$ -th element of a pipe-related variable  $\mathbf{x} \in \mathbb{R}^m$  would be denoted as  $x_{(i,j)}$  for  $i, j = 1, 2, \dots, n$  instead of  $x_k$  for  $k = 1, 2, \dots, m$  (thus, each pair  $(i, j)$  is directly related to an index  $k$ ). This improves the readability of the section by expressing all the formulas in terms of node indices.

calculations we arrive at the straight-forward relation  $\mathbf{B} = \frac{1}{2} (|\mathbf{\Lambda}|^T \mathbf{\Lambda} - \mathbf{\Lambda}^T |\mathbf{\Lambda}|)$ .  $\blacklozenge$

Considering that the balance of inflows and outflows passing throughout a node of index  $i$  can be expressed as:

$$\sum_{j: b_{ij} \neq 0} f_{(i,j)} = c_i, \quad (9)$$

where  $c_i$  is the consumption (the normal user demand and, potentially, leaks), by introducing (7) in (9), we arrive at

$$\sum_{j: b_{ij} > 0} \sigma_{(i,j)}^{0.54} (\psi_i - \psi_j)^{0.54} - \sum_{j: b_{ij} < 0} \sigma_{(i,j)}^{0.54} (\psi_j - \psi_i)^{0.54} = c_i, \quad (10)$$

or, equivalently stated,

$$\sum_{j: b_{ij} \neq 0} b_{ij} \sigma_{(i,j)}^{0.54} [b_{ij} (\psi_i - \psi_j)]^{0.54} = c_i. \quad (11)$$

Recall that head values in (11) corresponding to reservoirs are fixed. Without loss of generality, these known head values may be introduced into equations (11) such that only the ‘true’ variables, the junction node heads, remain. In any case, solving the group of equations (11) to retrieve the steady-state node heads is not an easy task. This is why hydraulic simulators like EPANET [34] are used, as they provide numerical solutions instead of attempting to derive analytic ones.

Thus, our goal is not to solve the WDN problem but rather to exploit the relations underlined by (11) to find first order (linear) dependencies between the current junction node’s head,  $\psi_i$ , and its neighbors,  $\psi_j$ . To do so, we apply the *implicit function theorem* [35], as detailed next.

**Lemma 1.** For the  $i$ -th node relation (11), assuming that there exist  $\bar{\psi}_i, \bar{\psi}_j, \forall j \in \mathcal{J} = \{j \mid b_{ij} \neq 0\}$  which verify it, there:

- i) exists a local (in the neighborhood of  $\bar{\psi}_i, \bar{\psi}_j$ ) and explicit dependence between its head ( $\psi_i$ ) and its neighbors’ heads ( $\psi_j, \forall j \in \mathcal{J}$ ),

$$\psi_i = \mathbf{g}_i(\boldsymbol{\psi}_{\mathcal{J}}), \quad (12)$$

with  $\boldsymbol{\psi}_{\mathcal{J}} = [\dots \psi_j \dots]^T \in \mathbb{R}^{|\mathcal{J}|}$  grouping all neighboring node’s head values;

- ii) and function’s  $\mathbf{g}_i : \mathbb{R}^{|\mathcal{J}|} \mapsto \mathbb{R}$  derivatives are given by<sup>3</sup>

$$\frac{\partial \mathbf{g}_i(\boldsymbol{\psi}_{\mathcal{J}})}{\partial \psi_j} = \frac{\sigma_{(i,j)}^{0.54} [b_{ij} (\psi_i - \psi_j)]^{-0.46}}{\sum_{k: b_{ik} \neq 0} \sigma_{ik}^{0.54} [b_{ik} (\psi_i - \psi_k)]^{-0.46}}, \quad (13)$$

for all  $j \in \mathcal{J}$ .  $\blacksquare$

*Proof.* See the appendix.  $\square$

**Proposition 1.** For the current node  $i$ , consider the head values  $\bar{\psi}_i, \bar{\psi}_j, \forall j \in \mathcal{J}$  which verify (11). Then,  $\mathbf{g}_i(\cdot)$ , defined as in (12) is approximated by the linear relation

$$\psi_i = \mathbf{g}_i(\boldsymbol{\psi}_{\mathcal{J}}) \approx \bar{\psi}_i + \sum_{j: b_{ij} \neq 0} \eta_{ij} \cdot (\psi_j - \bar{\psi}_j), \quad (14)$$

<sup>3</sup>Note that  $k$  is used here to index the neighbors of the  $i$ -th node during the summation, and it is not the same as the  $k$  used before to denote the  $k$ -th edge. This must be considered for subsequent analogue usages of  $k$  as an index.

where the weights  $\eta_{ij}$  are given by

$$\eta_{ij}(\bar{\psi}_i, \bar{\psi}_j) = \frac{\sigma_{(i,j)}^{0.54} [b_{ij}(\bar{\psi}_i - \bar{\psi}_j)]^{-0.46}}{\sum_{k: b_{ik} \neq 0} \sigma_{(i,j)}^{0.54} [b_{ik}(\bar{\psi}_i - \bar{\psi}_k)]^{-0.46}}, \quad (15)$$

for all  $j \in \mathcal{J}$ .  $\blacksquare$

*Proof.* As per the first part of Lemma 1, we have that there exists  $\mathbf{g}_i$  defined as in (12). We approximate this function by the first two terms of its Taylor expansion centered around  $\bar{\psi}_i$ ,  $\bar{\psi}_j, \forall j \in \mathcal{J}$ :

$$\mathbf{g}_i(\boldsymbol{\psi}_{\mathcal{J}}) \approx \bar{\psi}_i + \sum_{j: b_{ij} \neq 0} \left[ \frac{\partial \mathbf{g}_i(\boldsymbol{\psi}_{\mathcal{J}})}{\partial \psi_j} \Big|_{(\bar{\psi}_i, \bar{\psi}_j)} \cdot (\psi_j - \bar{\psi}_j) \right]. \quad (16)$$

Introducing relations (13) into (16) leads to:

$$\mathbf{g}_i(\boldsymbol{\psi}_{\mathcal{J}}) \approx \bar{\psi}_i + \frac{\sum_{j: b_{ij} \neq 0} \sigma_{(i,j)}^{0.54} [b_{ij}(\bar{\psi}_i - \bar{\psi}_j)]^{-0.46} \cdot (\psi_j - \bar{\psi}_j)}{\sum_{k: b_{ik} \neq 0} \sigma_{(i,j)}^{0.54} [b_{ik}(\bar{\psi}_i - \bar{\psi}_k)]^{-0.46}}. \quad (17)$$

Rearranging the terms of (17) directly leads to the values from (15), thus concluding the proof.  $\square$

### B. Graph-based state interpolation reformulation

The weights  $\eta_{ij}$  in (15) can be structured like those in (1),

$$\eta_{ij} = \frac{1}{\phi_i} \omega_{ij}, \quad (18)$$

where

$$\begin{aligned} \omega_{ij}(\bar{\psi}_i, \bar{\psi}_j) &= \sigma_{(i,j)}^{0.54} [b_{ij}(\bar{\psi}_i - \bar{\psi}_j)]^{-0.46}, \\ \phi_i(\bar{\psi}_i) &= \sum_{k: b_{ik} \neq 0} \sigma_{(i,j)}^{0.54} [b_{ik}(\bar{\psi}_i - \bar{\psi}_k)]^{-0.46}. \end{aligned} \quad (19)$$

Thus,  $\phi_i = \sum_{k: b_{ik} \neq 0} \omega_{ik}$  holds, letting us consider  $\omega_{ij}$  as the  $i-j$  entry of a weighted adjacency matrix  $\boldsymbol{\Omega}$  that encodes the relation between nodes  $v_i$  and  $v_j$ , with  $\phi_i$  as the degree of node  $v_i$ . Substituting (18) into (14), we obtain

$$\psi_i = \mathbf{g}_i(\boldsymbol{\psi}_{\mathcal{J}}) \approx \bar{\psi}_i + \frac{1}{\phi_i} \sum_{j: b_{ij} \neq 0} \omega_{ij} \cdot (\psi_j - \bar{\psi}_j), \quad (20)$$

which is written in matrix form as

$$\boldsymbol{\psi} - \bar{\boldsymbol{\psi}} \approx \boldsymbol{\Phi}^{-1} \boldsymbol{\Omega} (\boldsymbol{\psi} - \bar{\boldsymbol{\psi}}). \quad (21)$$

To arrive at an interpolation procedure similar with (4) we replace (21) with

$$\hat{\boldsymbol{\psi}} - \hat{\boldsymbol{\psi}} = \boldsymbol{\Delta} \hat{\boldsymbol{\psi}} = \boldsymbol{\Phi}^{-1} \boldsymbol{\Omega} (\hat{\boldsymbol{\psi}} - \hat{\boldsymbol{\psi}}). \quad (22)$$

which uses the estimated hydraulic head vector instead.

The leak-free estimated hydraulic heads are selected to perform the role of  $\hat{\boldsymbol{\psi}}$ , so that  $\boldsymbol{\Delta} \hat{\boldsymbol{\psi}}$  denotes the estimated pressure residuals. Thus, the minimization of the difference between the two sides of (22), which is analogue to (5), implies that the formulation proposed in (4) must be modified

to consider the change in the interpolated variable, i.e., the method interpolates pressure residuals instead of heads

$$\min_{\boldsymbol{\Delta} \hat{\boldsymbol{\psi}}} \frac{1}{2} [\boldsymbol{\Delta} \hat{\boldsymbol{\psi}}^T \mathbf{L} \boldsymbol{\Phi}^{-2} \mathbf{L} \boldsymbol{\Delta} \hat{\boldsymbol{\psi}}], \quad (23a)$$

$$\text{s.t. } \mathbf{Z} \boldsymbol{\Delta} \hat{\boldsymbol{\psi}} = \boldsymbol{\Delta} \hat{\boldsymbol{\psi}}_{\zeta}. \quad (23b)$$

Note that the directionality term of the optimization problem (4) no longer explicitly appears, as this direction cannot be imposed for the residuals. Nevertheless, it is indirectly included during the derivation of the weights (see (19)). The procedure of solving the problem posed in (23) is henceforward referred to as AW-GSI (analytical weights graph-based interpolation).

### C. Leak Localization via Dictionary Learning

We achieve leak localization via dictionary learning classification on the available interpolated data. The training phase of dictionary learning (DL) is performed on the samples generated from the application of AW-GSI to the available pressure residuals. First, we briefly overview the DL algorithms and their relation to fault isolation.

Sparse representations [36] are used to recover information from noisy samples through the use of a redundant basis, often called dictionary. For WDN, the samples are constituted by residuals from sensor readings and interpolated information, for various nominal and leaky scenarios taking place across the network nodes. Given  $n_{ts}$  total sensors (physical and virtual) and  $n_{samp}$  scenarios, we collect this data in the matrix  $\mathbf{Y} \in \mathbb{R}^{n_{ts} \times n_{samp}}$  with which we aim to recover the node where the fault took place. In our case,  $\mathbf{Y}$  would be filled with the entries  $\boldsymbol{\Delta} \hat{\boldsymbol{\psi}}$ , obtained from applying AW-GSI for all the considered leak scenarios and their corresponding data samples.

Given a dictionary  $\mathbf{D} \in \mathbb{R}^{n_{ts} \times n_{atom}}$  consisting of  $n_{atom}$  columns, also called atoms, we obtain the sparse representations  $\mathbf{X} \in \mathbb{R}^{n_{atom} \times n_{samp}}$ . Each column in  $\mathbf{X}$  is  $s$  sparse. To maximize precision, we specialize the dictionary for each WDN through the process of dictionary learning [37]. Here we are not solely interested in optimum representations, but also in finding the dictionary best fitted for the WDN at hand,

$$\begin{aligned} \min_{\mathbf{D}, \mathbf{X}} \quad & \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2, \\ \text{s.t.} \quad & \|\mathbf{x}_\ell\|_0 \leq s, \quad \ell = 1 : n_{samp}, \\ & \|\mathbf{d}_j\| = 1, \quad j = 1 : n_{atom}. \end{aligned} \quad (24)$$

where  $\|\cdot\|_0$  is the pseudo-norm counting the number of nonzeros. The goal is to obtain a specialized dictionary  $\mathbf{D}$  such that when we are given a new measurement  $\mathbf{y}$  we can closely approximate it as  $\mathbf{y} \approx \mathbf{D}\mathbf{x}$  through the use of a sparse representation algorithm. In this paper we will be using Orthogonal Matching Pursuit (OMP) [38] for this task.

Model-free identification of a leaky node from the set of network nodes is a classification task. To improve DL classification, we will be learning three separate dictionaries:  $\mathbf{D}$  to approximate the data,  $\mathbf{W}$  to classify the data, and  $\mathbf{A}$  to specialize atom blocks per each class. A class is a network node in our case. This approach is called Label Consistent K-

SVD (LC-KSVD) [39] and it extends (24) to simultaneously perform DL on all three dictionaries

$$\min_{D, \mathbf{W}, \mathbf{A}, \mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \alpha \|\mathbf{H} - \mathbf{W}\mathbf{X}\|_F^2 + \beta \|\mathbf{Q} - \mathbf{A}\mathbf{X}\|_F^2. \quad (25)$$

As it can be seen, each dictionary is trained on different data.  $\mathbf{H} \in \mathbb{R}^{n_{class} \times n_{samp}}$  represents the label matrix, where column  $h_i$  has 1 set in the class corresponding to the leaky node in measurements  $y_i$  and zeros elsewhere.  $\mathbf{Q} \in \mathbb{R}^{n_{atom} \times n_{samp}}$  enforces atom sets per class, where column  $q_i$  has ones in the positions corresponding to atoms that should represent the class to which  $y_i$  belongs and zeros elsewhere. Please note that in this process we start with  $\mathbf{Y}$  only containing sensor information, and add the labels  $\mathbf{H}$  to help identify non-sensorized nodes.

After the learning process, the dictionaries are used to perform fault detection and identification (FDI) on new sensor measurements  $\mathbf{y}$  in a two-step process: sparse representation  $\mathbf{y} \approx \mathbf{D}\mathbf{x}$  and classification  $i = \arg \max_j (\mathbf{W}\mathbf{x})_j$ , thus obtaining the leaky node  $i$ .

**Remark 4.** *Looking at computational complexity, the classification effort evolves around the sparse representation step where OMP uses  $O(sn_{ts}n_{atom})$  instructions for each data-item. Dictionary learning consists of  $K$  rounds of sparse representation and dictionary update. LC-KSVD uses  $O(sn_{ts}^2n_{samp})$  instructions to update the dictionary, sparse representation OMP performance over one data-item uses  $O(sn_{ts}n_{atom})$  operations, whereas the dictionary update in LC-KSVD has a complexity of  $O(sn_{ts}^2n_{samp})$ .* ♦

The choice in dictionary learning is motivated by our success in the past [25], [26], where DL was used without interpolation to tackle FDI. For large WDNs we extended this approach for online semi-supervised learning in [40].

#### IV. METHODOLOGY OVERVIEW

Once the core problem of AW-GSI has been defined, let us develop the operational flow of this leak localization methodology that combines interpolation and dictionary learning.

1) *Data generation:* The application of a supervised learning-based technique implies the necessity of sufficiently rich sets of samples and labels. In our case, the labels correspond to the different nodes of the network that may leak. Meanwhile, the samples are data vectors containing information about the state of the actual and virtual sensors at different scenarios and conditions. Both samples and labels are required from all possible leaks. To obtain this data, available historical leak datasets from water utilities can be useful. Besides, leak experiments can be executed over the field in order to gain data about the behaviour of the network on those abnormal events.<sup>4</sup> If these sources of information are not available, a hydraulic model of the network can be used to generate the necessary leak events.

Several aspects/parameters need to be taken into account about the data generation process:

<sup>4</sup>Note that nodes whose leaks cause similar effects over the network may be grouped into a common leak group, effectively reducing the number of labels in the learning phase, and hence the number of necessary experiments.

---

#### Algorithm 2 AW-GSI

---

**Require:**  $\hat{\psi}_\zeta, \hat{\psi}_\zeta \in \mathbb{R}^{n_\zeta}, \sigma \in \mathbb{R}^m$

- 1: Compute  $\hat{\psi}^{GSI}$  from  $\hat{\psi}_\zeta$  solving (4) [Remark 1]
  - 2: Compute  $\hat{\mathbf{B}}$  from  $\hat{\psi}^{GSI}$  using (8)
  - 3: Compute  $\Omega^{AW}, \Phi^{AW}$  from  $\hat{\mathbf{B}}, \sigma, \hat{\psi}^{GSI}$  using (19)
  - 4: Compute  $\mathbf{L}^{AW} = \Phi^{AW} - \Omega^{AW}$
  - 5: Compute  $\Delta\hat{\psi}_\zeta = \hat{\psi}_\zeta - \hat{\psi}_\zeta$
  - 6: Compute  $\Delta\hat{\psi}$  from  $\Delta\hat{\psi}_\zeta, \mathbf{L}^{AW}, \Phi^{AW}$  solving (23)
  - 7: **return**  $\Delta\hat{\psi}$
- 

- The number of considered leaks, i.e., different classes/labels,  $n_{class} \in \mathbb{N}$ .
- The number of time instants per leak,  $n_t \in \mathbb{N}$ .
- The number of considered leak sizes,  $n_{ls} \in \mathbb{N}$ .

Then, the complete leak dataset is going to be composed by  $n_{samp} = n_{class}n_t n_{ls}$  samples of length  $n_\zeta$ , i.e.,  $\hat{\Psi}_\zeta \in \mathbb{R}^{n_\zeta \times n_{samp}}$ . Additionally, note that the dataset must be divided into training and testing sets for the sake of the dictionary learning phase, i.e.,  $\hat{\Psi}^{train} \in \mathbb{R}^{n_\zeta \times n_{train}}$  and  $\hat{\Psi}^{test} \in \mathbb{R}^{n_\zeta \times n_{test}}$ , with  $n_{samp} = n_{train} + n_{test}$ .

**Remark 5.** *Apart from the leak dataset, a leak-free dataset is necessary because DL operates over residuals, which encode the difference in head state between nominal and leak conditions. Thus, each entry of the nominal set  $\hat{\Psi}_\zeta \in \mathbb{R}^{n_\zeta \times n_{samp}}$  must be obtained with network boundary conditions (e.g. water inlet pressure, nodal demands, etc.) similar to its analogue leak entry in  $\hat{\Psi}_\zeta$ , with the aim of minimizing possible state differences that are not related to the leak but to changes in consumer demands. The nominal dataset is also divided into training and testing, yielding  $\hat{\Psi}^{train} \in \mathbb{R}^{n_\zeta \times n_{train}}$  and  $\hat{\Psi}^{test} \in \mathbb{R}^{n_\zeta \times n_{test}}$ .* ♦

2) *AW-GSI:* The quadratic programming problem in (23) is the base of AW-GSI. Let us review the key points of the interpolation operation, summarized by Algorithm 2.

First, the selection of  $\hat{\psi}$  should be defined. Considering that the learning phase is fed with pressure residuals, a smart choice would be to select nominal/leak-free values to be represented by  $\hat{\psi}_\zeta$ . The standard GSI problem in (4) is solved to obtain  $\hat{\psi}^{GSI}$  from  $\hat{\psi}_\zeta$ , but considering the cost function stated in Remark 1, in order to explicitly pursue the state smoothness over the neighborhood. The complete nominal state vector is then used to compute both the approximate incidence matrix  $\hat{\mathbf{B}}$  and the physical-based method weights, i.e.,  $\Omega^{AW}$  and  $\Phi^{AW}$ , enabling the calculation of the Laplacian matrix  $\mathbf{L}^{AW}$ . Finally, the problem in (23) is solved, obtaining the desired interpolated residual vector  $\Delta\hat{\psi}$ .

3) *Selection of virtual sensors:* While more information should improve the leak localization performance, the disturbance induced in the learning phase by the interpolation errors becomes, at some point, significant. Thus, we propose to select a set  $\mathcal{Z}_v$  of  $n_{vs}$  nodes, which will append their interpolated state to that of the actual sensors. In this way, the total set of sensors would be composed of  $n_{ts} = n_\zeta + n_{vs}$  nodes. Please note that increasing the number of VSs means increasing  $n_{ts}$

---

**Algorithm 3** AW-GSI-DL — Training
 

---

**Require:**  $\hat{\Psi}_{\zeta}^{train}, \hat{\Psi}_{\zeta}^{train} \in \mathbb{R}^{n_{\zeta} \times n_{train}}, \sigma \in \mathbb{R}^m, \mathbf{H}^{train} \in \mathbb{R}^{n_{class} \times n_{train}}, s \in \mathbb{R}, n_{vs} \in \mathbb{R}$

- 1: Compute  $\Delta \hat{\Psi}^{train}$  from  $\hat{\Psi}_{\zeta}^{train}, \hat{\Psi}_{\zeta}^{train}, \sigma$  using Algorithm 2
  - 2: Select a set  $\mathcal{Z}_v$  of  $n_{vs}$  virtual sensors using [28]
  - 3: Extract  $\Delta \hat{\Psi}_{ts}^{train}$  from  $\Delta \hat{\Psi}^{train}$  using  $\mathcal{Z}_v$
  - 4: Compute  $\mathbf{D}, \mathbf{W}, \mathbf{A}$  from  $\Delta \hat{\Psi}_{ts}^{train}, \mathbf{H}^{train}$  solving (24)
  - 5: **return**  $\mathbf{D}, \mathbf{W}, \mathcal{Z}_v$
- 

**Algorithm 4** AW-GSI-DL — Application
 

---

**Require:**  $\hat{\Psi}_{\zeta}^{test}, \hat{\Psi}_{\zeta}^{test} \in \mathbb{R}^{n_{\zeta} \times n_{test}}, \sigma \in \mathbb{R}^m, s \in \mathbb{R}, \mathcal{Z}_v$

- 1: Compute  $\Delta \hat{\Psi}^{test}$  from  $\hat{\Psi}_{\zeta}^{test}, \hat{\Psi}_{\zeta}^{test}, \sigma$  using Algorithm 2
  - 2: Extract  $\Delta \hat{\Psi}_{ts}^{test}$  from  $\Delta \hat{\Psi}^{test}$  using  $\mathcal{Z}_v$
  - 3: Compute  $\mathbf{X}$  from  $\Delta \hat{\Psi}_{ts}^{test}, \mathbf{D}, s$  using OMP
  - 4: **return**  $i = \arg \max_{j=1:n_{class}} (\mathbf{W}\mathbf{x})_j, \forall \mathbf{x} \in \mathbf{X}$
- 

which, as per Remark 4, results in a linear increase in the FDI step and a quadratic increase in the learning process.

This selection is performed by means of a data-driven sensor placement methodology [28], in order to minimize the use of the hydraulic model. This approach pursues the sensor configuration that minimizes a certain topological-based metric (e.g. Section 2.3 and the explanations around (6) in [28]), related with the distance from the sensors to the rest of the nodes. Moreover, it allows to settle a set of pre-defined sensors, so that the rest of them are placed considering that the pre-defined ones are fixed, enabling virtual sensors to be selected in areas where the real sensor density is low. In this way, we can achieve the final training and testing datasets  $\Delta \hat{\Psi}_{ts}^{train} \in \mathbb{R}^{n_{ts} \times n_{train}}$  and  $\Delta \hat{\Psi}_{ts}^{test} \in \mathbb{R}^{n_{ts} \times n_{test}}$ .

4) *Learning/classification stages:* Once the input datasets to the learning stage are ready, the DL algorithm is trained and tested. Note that each entry of the datasets must be associated to its corresponding label, i.e., the leak scenario that originated the stored data. These matrices are referred to as  $\mathbf{H}^{train} \in \mathbb{R}^{n_{class} \times n_{train}}$  and  $\mathbf{H}^{test} \in \mathbb{R}^{n_{class} \times n_{test}}$ . The training and testing methodologies, describing the complete AW-GSI and DL processes, are detailed in Algorithms 3-4.

Note that the application algorithm can be operated in real-time, providing single head state vectors (for the sensorized nodes) instead of a testing dataset matrix. The nominal information can be retrieved from a historical dataset, or from the instants before the leak detection.

## V. RESULTS AND DISCUSSION<sup>5</sup>

The proposed methodology, AW-GSI-DL, is implemented and tested over a case study network, comparing its performance in terms of interpolation accuracy and localization results with respect to GSI-DL [27]. Moreover, both methods are compared with a purely data-driven approach, GSI-LCSM [19]. Thus, the advantages of the presented updates to the interpolation process are highlighted, as well as the improvements

<sup>5</sup>Code and data available at <https://github.com/pirofti/AW-GSI-DL>

TABLE I  
MODENA WDN CHARACTERISTICS

Property	Value
Number of inner junctions	268
Number of pipes	317
Number of reservoirs (water inlets)	4
Total pipe length	71.8 km
Total nodal demand	~400 l/s

of including the learning stage to help the interpolation scheme with the localization task. All the localization simulations were performed on an AMD Ryzen Threadripper PRO 3955WX with 512GB of memory using Octave 8.1.0.

### A. Case study

The network of Modena, Italy, constitutes a well-known case study in the WDNs management field [12], [21], [41]. It is an open-source benchmark, which helps with the replicability and comparison of results. The network topology is schematically represented in Fig. 1, and its main properties are summarized by Table I. Both the network size (in terms of number of nodes/pipes, reservoirs and pipe lengths) and the total demand correspond to a problem of realistic size.

A set of 20 pressure sensors are considered to be deployed over the network, i.e., 4 sensors at the water inlets and 16 at inner nodes. The sensor locations are decided by means of the previously mentioned data-driven sensor placement technique, presented in [28].

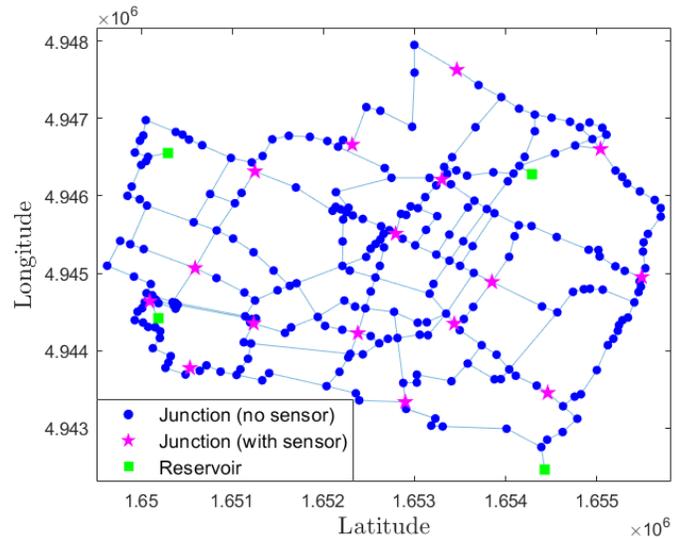


Fig. 1. Schematic representation of the Modena WDN.

### B. Data generation

Hydraulic data from a wide set of leak scenarios is required to assess the performance of the methodology. To this end, EPANET has been used to generate leak data for all the

possible scenarios regarding the leak location. The values of the data generation parameters presented in Section IV-1 in the performed EPANET simulations are the following:

- All the possible leaks are considered, i.e.,  $n_{class} = 268$  (excluding the 4 water inlets).
- The simulations cover a period of 24 hours, with a timestep of 1 hour, and hence  $n_t = 24$ .
- Several leak sizes have been considered to replicate realistic scenarios. In this case, training sets have been derived considering four different leak sizes, ranging from 4 l/s to 7 l/s (i.e.,  $\sim 1-1.75\%$  of the average total inflow) with a step of 1 l/s. Three testing leak sizes have been selected to be different to the training ones but within the training range: 4.5, 5.5 and 6.5 l/s. Thus,  $n_{ls} = 7$ . Using leaks with smaller magnitudes for training makes the FDI mechanism more challenging since the discrimination between “healthy but affected by uncertainties” and “under fault” becomes harder to check. Thus, we consider the current leak range to be adequate for both training and subsequent testing. Not least, recent studies dealing with the same benchmark consider similar or even larger leak sizes because of this fact [21].

Additionally, several sources of uncertainty have been considered during the data generation to test the approach in realistic conditions. On the one hand, sensor precision is considered, trusting the generated head values until a precision of  $\pm 1$  cm. Thus, leak effects are not noticeable until the minimum level of precision is reached. On the other hand, uniformly random uncertainty has been added to physical properties of the WDN, specifically pipe diameters and roughness, as well as the daily demand patterns. Two cases have been studied: 0.5% and 1% of uncertainty with respect to the noise-free value. Note that the methodology requires a leak-free scenario, which is also affected by uncertainty, to achieve the pressure residuals. Thus, the input data to the learning stage is affected by a higher degree of uncertainty.

Let us remark that considering all the mentioned sources of uncertainty, i.e., leak size, modeling (network) properties, sensor precision and demand pattern, is not common among most of the state-of-the-art mixed model-based/data-driven leak detection/localization methods, as it can be appreciated in Table I of [8], which summarizes the features of 54 relevant articles in the field.

Finally, let us remark that leaks are generated by means of the emitter component available in EPANET. It models the flow through an orifice that leaks to the environment as

$$f_i^{leak} = \epsilon \rho_i^{0.5}, \quad (26)$$

where  $\epsilon$  is the emitter coefficient, and  $\rho_i$  is the pressure at node  $v_i$ .

### C. Interpolation performance

In order to analyze the improvements of AW-GSI with respect to GSI in terms of state estimation performance, let us derive two different metrics related to the difference between interpolated and actual values. Specifically, they are based on the root-squared-mean error (RMSE), which is computed as

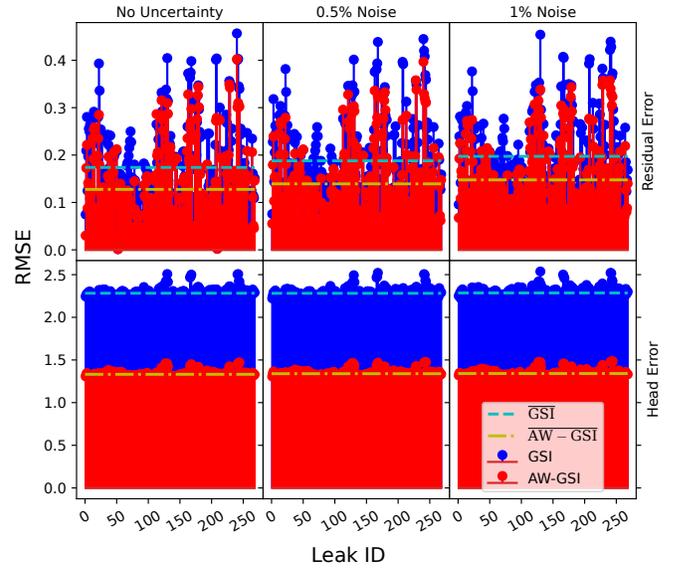


Fig. 2. Interpolation RMSE per leak – GSI vs AW-GSI – head and residual errors.

$$RMSE(\mathbf{x}, \hat{\mathbf{x}}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2}, \quad (27)$$

where  $\mathbf{x}$  is a generic vector whose  $i$ -th element corresponds to a value at node  $v_i$ , and  $\hat{\mathbf{x}}$  is the computed estimation of  $\mathbf{x}$ .

The first metric assesses the accuracy of the hydraulic head reconstruction by calculating the error between actual EPANET and interpolated values, i.e.,  $RMSE(\psi, \hat{\psi})$ . The second metric, computed as  $RMSE(\Delta\psi, \Delta\hat{\psi})$ , gauges the similarity between the residuals received by the learning scheme from interpolation and EPANET, indicating the quality of the training data. Both calculations are performed for each leak case and time instant, averaging over the latter to obtain a RMSE value for each leak.

Note again that GSI retrieves heads and AW-GSI yields residuals. Thus, for the sake of the comparisons, it is required to compute the residuals associated to the GSI heads and the heads that correspond to the AW-GSI residuals. For GSI, the difference between the each leak head vector and the corresponding leak-free data yields the GSI associated residuals. For AW-GSI, the nominal heads vector is added to the estimated residuals to obtain the leak hydraulic heads.

The interpolation results are displayed in Fig. 2. The figure is divided into three columns and two rows. Each column represents a different scenario regarding the considered uncertainty. Specifically, the cases of 0, 0.5 and 1% of uncertainty are exposed. Regarding the rows, the first one shows the results for the residual-error, whereas the second row shows the head-reconstruction error. Each vertical line denotes the RMSE of the corresponding leak, with blue and red lines showing GSI and AW-GSI performance respectively. Note that the mean values (over the leaks) are indicated with cyan and green horizontal lines for GSI and AW-GSI respectively. Important conclusions can be extracted from these results:

- About the head-reconstruction error, AW-GSI clearly outperforms GSI, yielding a lower error for all the possible leaks. An analysis of the mean values (over the leaks) reveals a reduction of 41.65% in the head-error. Thus, we conclude that AW-GSI is clearly capable of estimating the network hydraulic heads with a higher degree of accuracy.
- Regarding the residual-error, the difference in terms of performance between both interpolation schemes is not as large as in the previous case. Even so, the error associated to AW-GSI is lower than the error of GSI in a 88.06% of the leak scenarios, with an average (over the leaks) reduction of 26.62% in RMSE. Thus, AW-GSI arguably performs better than GSI regarding residual-error too.
- Analyzing the effect of uncertainty, it becomes clear that the interpolation process is not degraded. Both GSI and AW-GSI are robust against physical properties and demand pattern noise. This is specially interesting for AW-GSI, considering that the AW-generation is fed with the noise-free pipe diameters and roughness, and hence the weights are obtained with information that slightly differs from the one used to generate the hydraulic data.

#### D. Localization performance

The interpolation results from the previous stage, for both GSI and AW-GSI, were used as inputs to the DL process, leading to the training and posterior testing of the methodologies for leak localization. Note that DL is fed with normalized residuals, so they must be computed in the case of GSI from the interpolated leak and leak-free vectors (both GSI and AW-GSI residuals have to be normalized).

The localization results are presented at Fig. 3. It is divided into three columns, each displaying performance for different uncertainty levels (0%, 0.5%, and 1%). In each column, three accuracy levels are compared based on the number of successful targets considered:

- Node-level accuracy (indicated by circle markers): indicates the percentage of testing samples for which the exact leaky node is selected by the trained DL scheme.
- 1-neighbour level accuracy (indicated by triangle markers): denotes the percentage of successful tests when the 1st layer of neighbours of the exact leaky node are considered as correct locations too.
- 2-neighbour level accuracy (indicated by square markers): it is analogue to the previous metric, but considering the exact node and the first and second layers of neighbours.

Besides, the x-axis stands for the number of virtual sensors that are considered in each case. In this way, an analysis of the effect of including extra VS can be performed. Specifically, the cases of 0, 10, 20, 50, 100, 150, 200 and 252 are included (this last case implies that all the non-sensorized nodes provide information to the learning process through the interpolation data). Note that the previously mentioned sensor placement methodology [28] is used to add a set of new sensors (virtual) to the existing set of installed ones. In this way, the VS are located at areas with a low density of pressure sensors.

These results can be analyzed from various perspectives:

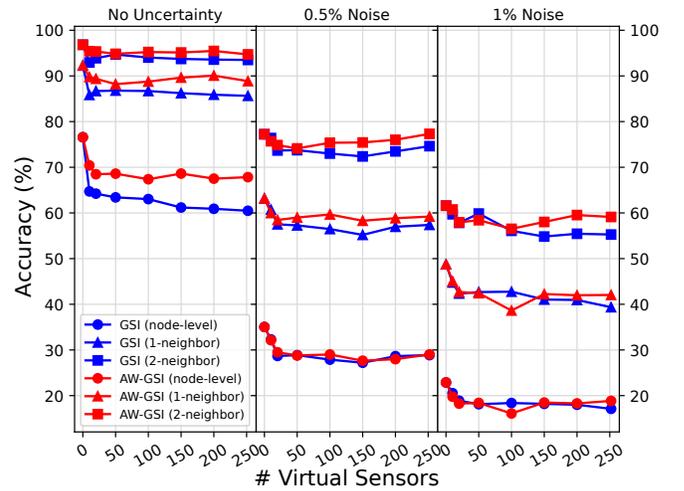


Fig. 3. Localization performance — GSI vs AW-GSI. The x-axis shows the number of VS considered, whereas the y-axis shows the localization accuracy (%). The three columns of subfigures represent different uncertainty levels: no uncertainty, 0.5% and 1%.

TABLE II  
EXECUTION TIMES FOR THE LOCALIZATION RESULTS PRESENTED IN FIG. 3.

#VS	0	50	100	150	252
t(min)	73 ± 0.7	74 ± 0.9	75 ± 1	76 ± 0.8	77 ± 0.7

- Regarding the comparison between both interpolation methodologies, it is clear that the localization accuracy is greater when AW-GSI is used instead of GSI, for the vast majority of tested scenarios.
- The inclusion of uncertainty degrades the localization performance, affecting both interpolation schemes. Nevertheless, the accuracy results for both 1 and 2-level neighbors are mostly satisfactory, considering that, in average, the 1-level layer of neighbors only occupies a 1.25% of the total area of network (considering the proportion of neighbors by total WDN nodes), whereas the 2-level layer of neighbors occupies 2.55% of the area.
- Note that the 2-neighbors level accuracy is closer between GSI and AW-GSI when no uncertainty is considered, while greater differences occur for the node-level case. However, the opposite is remarked when uncertainty is introduced. This occurs because uncertainty mostly hinders node-level accuracy, making node-level successful localizations with no uncertainty to fall to the category of 1 or 2-neighbors level successful localizations.
- Regarding the number of VSSs, it can be observed that the accuracy drops in comparison to the no-VS case. This is caused by the interpolation still introducing defects in the input data to the interpolation process. When starting to introduce VSSs, a sudden drop in the accuracy occurs, mostly in the node-level, due to the mismatch that may occur between measured and interpolated data. This drop is diminished or even overcome, as the extra information introduced when including more and more VS is consistent with the already provided virtual estimations.

However, let us remark that AW-GSI outperforms GSI in terms of diminishing the performance drop, even achieving higher performances for 252 VS than for more reduced sets of VS in some scenarios.

To complement the localization performance results, we also present in Table II the average execution times of the training process as the number of virtual sensors increase. We observe a slight increase as the number grows, but it is not significant or in any way prohibiting the use of numerous virtual sensors. Profiling showed that most time is spent during the sparse representation stage (79%), followed by the dictionary update (19%) and the rest is spent iterating over the two.

Another experiment has been carried out to highlight the contribution of the combination of interpolation and learning to increase the localization accuracy. Both interpolation-learning strategies, i.e., GSI-DL and AW-GSI-DL, are compared with the localization strategy, based on a leak candidate selection method (LCSM), which originally exploited GSI to interpolate approximated hydraulic heads: GSI-LCSM [19].

The graphical result of this comparison is presented in Fig. 4. In this figure, the x-axis stands for the considered depth of neighbours of the actual leak that are considered as successful targets for the classification process (as detailed for the cases of node-level, 1-neighbour level and 2-neighbour level accuracy in the explanations of Fig. 3). This depth is referred to as the localization area, as it correspond to the set or area of nodes that constitute a correct localization. Thus, the aim of this experiment is to study the evolution of the accuracy from node-level localization up to a certain degree of neighbours. In this case, a maximum of 6-neighbour level is considered, which represents a 13.31% of the total number of network nodes (in average). This selection is justified by the original nature of GSI-LCSM: the candidate selection stage of this method dynamically selects the size of the set of candidates (which can be regarded as the set of successful targets), as the methodology was designed to yield localization areas. For the sake of the comparison between GSI-LCSM and the two DL-based strategies, this dynamic set selection was substituted by the predefined sets of neighbours corresponding to each node and depth level. Thus, we considered appropriate to reach at least a 12-13% of the network with the maximum level, regarding the results obtained in terms of candidate areas in previous works based on GSI-LCSM [42]. Regarding GSI-DL and AW-GSI-DL, let us remark that the case of 252 VS (that is, all the network nodes are real or virtual sensors) is selected for the comparison, considering that GSI-LCSM exploits all the interpolated information during LCSM.

The results presented yield several interesting conclusions:

- The DL-based methods greatly improve accuracy at node-level (and reduced depths of neighbors) in comparison to GSI-LCSM. This confirms the suitability of the approach to accomplish one of the aims we were pursuing, specifically the improvement of the usually unreliable localization of purely data-driven methods at node-level.
- Although the performance of the three methods become more similar for greater depths of neighbors, the DL-based approaches continues yielding higher accuracy.

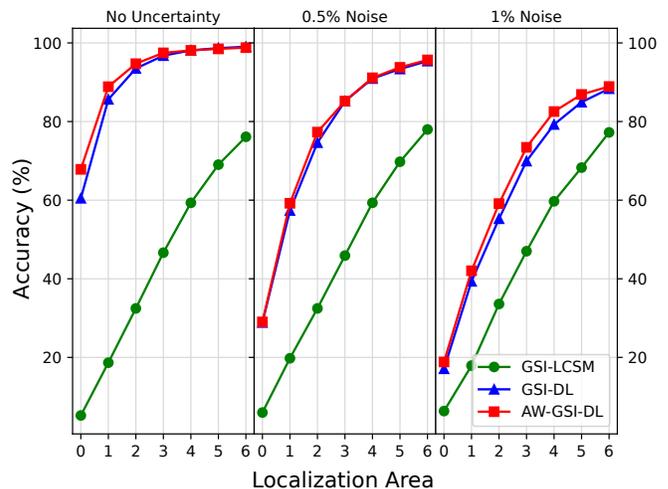


Fig. 4. Localization performance — GSI-LCSM vs GSI-DL vs AW-GSI-DL. The x-axis denotes the depth of neighbors of the leaky node considered as correct targets, and the y-axis represent the localization accuracy (%). From top to bottom, the subplots represent the cases considering 0%, 0.5% and 1% noise respectively.

- Noise degrades the node-level (and reduced depths of neighbors) accuracy of the learning-based techniques, although the performance still improves GSI-LCSM, whose results are not affected by noise.

## VI. CONCLUSIONS

This article presents a leak localization method combining interpolation and dictionary learning. The novel interpolation technique, i.e., AW-GSI, leverages the physics behind the relation between hydraulic heads of neighboring nodes in WDNs. Additionally, residuals are directly interpolated instead of hydraulic head values.

The method is applied to the well-known Modena case study, proving AW-GSI's improvements over GSI in terms of interpolation error, considering state and residual estimation; and posterior localization. Performance is tested for several uncertainty levels to assess its effect over the estimated heads/residuals and localization results. We note that uncertainty does not affect the interpolation process, although it degrades the localization accuracy up to some extent. Besides, localization experiments involved multiple virtual sensor sets and various degrees of target node depth.

Improvements are necessary to enhance the interpolation process and enrich the information for the learning stage. Additionally, boosting dictionary learning's robustness to noisy data can mitigate classification degradation caused by uncertainty. This will allow to overcome the limitations of the learning approaches caused by the noise, leading to a useful tool for water utilities to work with real data. Nevertheless, promising results have been obtained in this work, showing improvements with respect to previous methods, and confirming the suitability of the combined interpolation-learning scheme to compensate the weaknesses of interpolation-only methods.

APPENDIX  
PROOF OF LEMMA 1

The implicit function theorem states that for a function  $\mathbf{f} : \mathbb{R}^{n+m} \mapsto \mathbb{R}^m$  which respects the implicit relation  $\mathbf{f}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ , with  $\mathbf{x} \in \mathbb{R}^m$ ,  $\mathbf{y} \in \mathbb{R}^n$ , in a neighborhood around  $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$  verifying  $\mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = \mathbf{0}$ , there exists (under reasonable assumptions, [43]) an explicit dependence  $\mathbf{y} = \mathbf{g}(\mathbf{x})$  such that  $\mathbf{f}(\mathbf{x}, \mathbf{g}(\mathbf{x})) = \mathbf{0}$ . Moreover, we have that

$$\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}_j} = - \left[ \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}} \right]^{-1} \cdot \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x}_j}, \quad \forall j = 1 \dots n. \quad (28)$$

Equation (11) defines an implicit relation between the current junction's head  $\psi_i$  and its neighboring junctions' heads  $\{\psi_j\}_{j: b_{ij} \neq 0}$ . We may then define the implicit function  $\mathbf{f}_i(\psi_i, \boldsymbol{\psi}_{\mathcal{J}}) = 0$  where  $\mathbf{f}_i : \mathbb{R}^{|\mathcal{J}|+1} \mapsto \mathbb{R}$  and which comes from moving  $\mathbf{c}_i$  to the left of the equal sign in (11) and by grouping all neighboring node's head values in the vector  $\boldsymbol{\psi}_{\mathcal{J}}$ , defined as below (12). Note that the construction is particular in the sense that  $\mathbf{f}_i$  is a scalar function and thus the Jacobian inversion from (28) is a simple scalar division.

Applying the apparatus of the implicit function theorem we have that for a pair  $(\bar{\psi}_i, \bar{\boldsymbol{\psi}}_{\mathcal{J}})$  verifying  $\mathbf{f}_i(\bar{\psi}_i, \bar{\boldsymbol{\psi}}_{\mathcal{J}}) = 0$  there exists the scalar function  $\mathbf{g}_i : \mathbb{R}^{|\mathcal{J}|} \mapsto \mathbb{R}$  which verifies  $\psi_i = \mathbf{g}_i(\boldsymbol{\psi}_{\mathcal{J}})$  on a neighborhood of  $(\bar{\psi}_i, \bar{\boldsymbol{\psi}}_{\mathcal{J}})$ , thus proving (12) and allowing to adapt (28) into

$$\frac{\partial \mathbf{f}_i(\boldsymbol{\psi}_{\mathcal{J}})}{\partial \psi_j} = - \left[ \frac{\partial \mathbf{f}_i(\psi_i, \boldsymbol{\psi}_{\mathcal{J}})}{\partial \psi_i} \right]^{-1} \cdot \frac{\partial \mathbf{f}_i(\psi_i, \boldsymbol{\psi}_{\mathcal{J}})}{\partial \psi_j}, \quad \forall j \in \mathcal{J}. \quad (29)$$

Differentiating after  $\psi_i$  in (11), we obtain

$$\begin{aligned} \frac{\partial \mathbf{f}_i(\psi_i, \boldsymbol{\psi}_{\mathcal{J}})}{\partial \psi_i} &= \frac{\partial}{\partial \psi_i} \left( \sum_{k: b_{ik} \neq 0} b_{ik} \sigma_{(i,j)}^{0.54} [b_{ik}(\psi_i - \psi_k)]^{0.54} \mathbf{c}_i \right), \\ &= 0.54 \sum_{k: b_{ik} \neq 0} b_{ik}^2 \sigma_{(i,j)}^{0.54} [b_{ik}(\psi_i - \psi_k)]^{-0.46} \end{aligned} \quad (30)$$

We repeat the procedure for each  $\psi_j, j \in \mathcal{J}$  and obtain

$$\frac{\partial \mathbf{f}_i(\psi_i, \boldsymbol{\psi}_{\mathcal{J}})}{\partial \psi_j} = -0.54 b_{ij}^2 \sigma_{(i,j)}^{0.54} [b_{ij}(\psi_i - \psi_j)]^{-0.46}. \quad (31)$$

Noting that  $b_{ij}^2 = 1, \forall b_{ij} \neq 0$ , we introduce (30) and (31) into (29), thus arriving at (13) and concluding the proof.

ACKNOWLEDGMENT

The authors would like to thank the Spanish national project L-BEST (Ref. PID2020-115905RB-C21). This work was also supported by a grant of the Ministry of Research, Innovation and Digitization, CCCDI - UEFISCDI, project number PN-III-P2-2.1-PED-2021-1626, within PNCIDI III.

REFERENCES

[1] Mesfin M Mekonnen and Arjen Y Hoekstra, "Four billion people facing severe water scarcity," *Science advances*, vol. 2, no. 2, pp. e1500323, 2016, <https://doi.org/10.1126/sciadv.1500323>.  
[2] Barney Cohen, "Urban growth in developing countries: a review of current trends and a caution regarding existing forecasts," *World development*, vol. 32, no. 1, pp. 23–51, 2004, <https://doi.org/10.1016/j.worlddev.2003.04.008>.

[3] R. Liemberger and A. Wyatt, "Quantifying the global non-revenue water problem," *Water Supply*, vol. 19, no. 3, pp. 831–837, 2019, <https://doi.org/10.2166/ws.2018.129>.  
[4] Q. Xu, R. Liu, Q. Chen, and R. Li, "Review on water leakage control in distribution networks and the associated environmental benefits," *J. Environ. Sci.*, vol. 26, no. 5, pp. 955–961, 2014, [https://doi.org/10.1016/S1001-0742\(13\)60569-0](https://doi.org/10.1016/S1001-0742(13)60569-0).  
[5] M.W. LeChevallier, R.W. Gullick, M.R. Karim, M. Friedman, and J.E. Funk, "The potential for health risks from intrusion of contaminants into the distribution system from pressure transients," *J. Water Health*, vol. 1, no. 1, pp. 3–14, 2003, <https://doi.org/10.2166/wh.2003.0002>.  
[6] MJH Sterling and A Bargiela, "Leakage reduction by optimised control of valves in water networks," *Transactions of the Institute of Measurement and Control*, vol. 6, no. 6, pp. 293–298, 1984, <https://doi.org/10.1177/014233128400600603>.  
[7] Ranko S Pudar and James A Liggett, "Leaks in pipe networks," *Journal of Hydraulic Engineering*, vol. 118, no. 7, pp. 1031–1046, 1992, [https://doi.org/10.1061/\(ASCE\)0733-9429\(1992\)118:7\(1031\)](https://doi.org/10.1061/(ASCE)0733-9429(1992)118:7(1031)).  
[8] Luis Romero-Ben, Débora Alves, Joaquim Blesa, Gabriela Cembrano, Vicenç Puig, and Eric Duviella, "Leak detection and localization in water distribution networks: Review and perspective," *Annual Reviews in Control*, 2023, <https://doi.org/10.1016/j.arcontrol.2023.03.012>.  
[9] R. Perez, G. Sanz, V. Puig, J. Quevedo, M.A.C. Escofet, F. Nejjari, J. Meseguer, G. Cembrano, J.M.M. Tur, and R. Sarrate, "Leak localization in water networks: a model-based methodology using pressure sensors applied to a real network in barcelona [applications of control]," *IEEE Control Systems*, vol. 34, no. 4, pp. 24–36, 2014, <https://doi.org/10.1109/MCS.2014.2320336>.  
[10] David B Steffelbauer, Jochen Deuerlein, Denis Gilbert, Edo Abraham, and Olivier Pillier, "Pressure-leak duality for leak detection and localization in water distribution systems," *Journal of Water Resources Planning and Management*, vol. 148, no. 3, pp. 04021106, 2022, [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0001515](https://doi.org/10.1061/(ASCE)WR.1943-5452.0001515).  
[11] Zheng Yi Wu, Paul Sage, and David Turtle, "Pressure-dependent leak detection model and its application to a district water system," *Journal of Water Resources Planning and Management*, vol. 136, no. 1, pp. 116–128, 2010, [https://doi.org/10.1061/\(ASCE\)0733-9496\(2010\)136:1\(116\)](https://doi.org/10.1061/(ASCE)0733-9496(2010)136:1(116)).  
[12] Marcos Quiñones-Grueiro, Marlon Ares Milián, Maibeth Sánchez Rivero, Antônio J Silva Neto, and Orestes Llanes-Santiago, "Robust leak localization in water distribution networks using computational intelligence," *Neurocomputing*, vol. 438, pp. 195–208, 2021, <https://doi.org/10.1016/j.neucom.2020.04.159>.  
[13] F Costanzo, A Fiorini Morosini, P Veltri, and D Savić, "Model calibration as a tool for leakage identification in wds: A real case study," *Procedia Engineering*, vol. 89, pp. 672–678, 2014, <https://doi.org/10.1016/j.proeng.2014.11.493>.  
[14] G. Sanz, R. Pérez, and A. Escobet, "Leakage localization in water networks using fuzzy logic," *2012 20th Mediterranean Conference on Control & Automation (MED)*, pp. 646–651, 2012, <https://doi.org/10.1109/MED.2012.6265711>.  
[15] Y. Kim, S.J. Lee, T. Park, G. Lee, J.C. Suh, and J.M. Lee, "Robust leak detection and its localization using interval estimation for water distribution network," *Comput. Chem. Eng.*, vol. 92, pp. 1–17, 2016, <https://doi.org/10.1016/j.compchemeng.2016.04.027>.  
[16] A. Menapace, D. Avesani, M. Righetti, A. Bellin, and G. Pisaturo, "Uniformly distributed demand EPANET extension," *Water Resour. Manag.*, vol. 32, no. 6, pp. 2165–2180, 2018, <https://doi.org/10.1007/s11269-018-1924-6>.  
[17] J. Blesa and R. Pérez, "Modelling uncertainty for leak localization in water networks," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 730–735, 2018, <https://doi.org/10.1016/j.ifacol.2018.09.656>.  
[18] Adria Soldevila, Joaquim Blesa, Tom Nørgaard Jensen, Sebastian Tornil-Sin, Rosa M Fernández-Cantí, and Vicenç Puig, "Leak localization method for water-distribution networks using a data-driven model and Dempster–Shafer reasoning," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 937–948, 2020, <https://doi.org/10.1109/TCST.2020.2982349>.  
[19] Luis Romero-Ben, Débora Alves, Joaquim Blesa, Gabriela Cembrano, Vicenç Puig, and Eric Duviella, "Leak localization in water distribution networks using data-driven and model-based approaches," *Journal of Water Resources Planning and Management*, vol. 148, no. 5, pp. 04022016, 2022, [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0001542](https://doi.org/10.1061/(ASCE)WR.1943-5452.0001542).  
[20] A. Rajeswaran, S. Narasimhan, and S. Narasimhan, "A graph partitioning algorithm for leak detection in water distribution networks," *Computers & Chemical Engineering*, 2017, <https://doi.org/10.1016/j.compchemeng.2017.08.007>.

- [21] Débora Alves, Joaquim Blesa, Eric Duviella, and Lala Rajaoarisoa, “Data-driven leak localization in wdn using pressure sensor and hydraulic information,” *IFAC-PapersOnLine*, vol. 55, no. 5, pp. 96–101, 2022, <https://doi.org/10.1016/j.ifacol.2022.07.646>.
- [22] Miguel Capelo, Bruno Brentan, Laura Monteiro, and Dídia Covas, “Near-real time burst location and sizing in water distribution systems using artificial neural networks,” *Water*, vol. 13, no. 13, pp. 1841, 2021, <https://doi.org/10.3390/w13131841>.
- [23] Marlon Jesús Ares-Milián, Marcos Quiñones-Grueiro, Cristina Verde, and Orestes Llanes-Santiago, “A leak zone location approach in water distribution networks combining data-driven and model-based methods,” *Water*, vol. 13, no. 20, pp. 2924, 2021, <https://doi.org/10.3390/w13202924>.
- [24] Luis Romero, Joaquim Blesa, Vicenç Puig, and Gabriela Cembrano, “Clustering-learning approach to the localization of leaks in water distribution networks,” *Journal of Water Resources Planning and Management*, vol. 148, no. 4, pp. 04022003, 2022, [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0001527](https://doi.org/10.1061/(ASCE)WR.1943-5452.0001527).
- [25] P. Irofti and F. Stoican, “Dictionary learning strategies for sensor placement and leakage isolation in water networks,” in *The 20th World Congress of the International Federation of Automatic Control*, 2017, pp. 1589–1594, <https://doi.org/10.1016/j.ifacol.2017.08.308>.
- [26] P. Irofti, F. Stoican, and V. Puig, “Fault handling in large water networks with online dictionary learning,” *Journal of Process Control*, vol. 94, pp. 46–57, 2020, <https://doi.org/10.1016/j.jprocont.2020.08.003>.
- [27] P. Irofti, L. Romero-Ben, F. Stoican, and V. Puig, “Data-driven leak localization in water distribution networks via dictionary learning and graph-based interpolation,” in *Proceedings 2022 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2012, pp. 1265–1270, <https://doi.org/10.1109/CCTA49430.2022.9966160>.
- [28] Luis Romero-Ben, Gabriela Cembrano, Vicenç Puig, and Joaquim Blesa, “Model-free sensor placement for water distribution networks using genetic algorithms and clustering,” *IFAC-PapersOnLine*, vol. 55, no. 33, pp. 54–59, 2022, <https://doi.org/10.1016/j.ifacol.2022.11.009>.
- [29] A. Soldevila, J. Blesa, T. N. Jensen, S. Tornil-Sin, R. M. Fernández-Cantí, and V. Puig, “Leak localization method for water-distribution networks using a data-driven model and dempster-shafer reasoning,” *IEEE Transactions on Control Systems Technology*, pp. 937–948, 2021.
- [30] Saud A Taher and John W Labadie, “Optimal design of water-distribution networks with gis,” *Journal of Water Resources Planning and Management*, vol. 122, no. 4, pp. 301–311, 1996, [https://doi.org/10.1061/\(ASCE\)0733-9496\(1996\)122:4\(301\)](https://doi.org/10.1061/(ASCE)0733-9496(1996)122:4(301)).
- [31] Gerard Sanz Estapé, *Demand modeling for water networks calibration and leak localization*, Ph.D. thesis, Universitat Politècnica de Catalunya, 2016.
- [32] Jing-Chao Chen, “Dijkstra’s shortest path algorithm,” *Journal of formalized mathematics*, vol. 15, no. 9, pp. 237–247, 2003, <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=2dd9468be1a0ee733a3fb11008d4b2b2dba9e70>.
- [33] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty, “Semi-supervised learning using gaussian fields and harmonic functions,” in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 2003, pp. 912–919, <https://mlg.eng.cam.ac.uk/zoubin/papers/zgl.pdf>.
- [34] L. A. Rossman, *EPANET 2 User’s Manual*, Washington, D.C., 2000.
- [35] Raghavan Narasimhan, *Analysis on real and complex manifolds*, Elsevier, 1985.
- [36] M. Elad, *Sparse and Redundant Representations: from Theory to Applications in Signal Processing*, Springer, 2010.
- [37] B. Dumitrescu and P. Irofti, *Dictionary Learning Algorithms and Applications*, Springer, 2018.
- [38] Y.C. Pati, R. Rezaifar, and P.S. Krishnaprasad, “Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition,” in *27th Asilomar Conf. Signals Systems Computers*, Nov. 1993, vol. 1, pp. 40–44.
- [39] Z. Jiang, Z. Lin, and L.S. Davis, “Label Consistent K-SVD: Learning a Discriminative Dictionary for Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, 11 2013.
- [40] P. Irofti and A. Băltou, “Malware identification with dictionary learning,” in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.
- [41] Cristiana Bragalli, Claudia D’Ambrosio, Jon Lee, Andrea Lodi, and Paolo Toth, “On the optimal design of water distribution networks: a practical MINLP approach,” *Optimization and Engineering*, vol. 13, pp. 219–246, 2012, <https://doi.org/10.1007/s11081-011-9141-7>.
- [42] L. Romero, V. Puig, G. Cembrano, J. Blesa, and J. Meseguer, “A fully data-driven approach for leak localization in water distribution networks,” in *2021 European Control Conference (ECC)*, 2021, pp. 1851–1856, <https://doi.org/10.23919/ECC54610.2021.9654928>.
- [43] Murray H Protter and Charles B Morrey Jr, “Implicit function theorems. jacobians,” in *Intermediate Calculus*, pp. 390–420. Springer, 1985.



**Paul Irofti** is an Associate Professor within the Computer Science Department of the Faculty of Mathematics and Computer Science at the University of Bucharest and the Vice-President of the Institute for Logic and Data Science. He is the co-author of the book “Dictionary Learning Algorithms and Applications” (Springer 2018) awarded by the Romanian Academy. He is PhD in Systems Engineering at the Politehnica University of Bucharest since 2016. His interests are anomaly detection, signal processing, numerical algorithms and optimization.



**Luis Romero-Ben** is currently a PhD student and researcher at the Automatic Control group at the Institut de Robòtica e Informàtica Industrial (CSIC-UPC), Spain. His interests are focused on the development and application of data-driven methodologies for the control and monitoring of water distribution networks and urban drainage systems.



**Florin Stoican** is Professor in the department of Automatic Control and Systems Engineering, Politehnica University of Bucharest. He obtained his PhD in Control Engineering in 2011 from Supelec (now CentraleSupélec), France with an application of set-theoretic methods for fault detection and isolation. His interests are constrained optimization control, set theoretic methods, fault tolerant control, mixed integer programming, motion planning.



**Vicenç Puig** holds a PhD in Automatic Control, Vision and Robotics and is the leader of the research group Advanced Control Systems (SAC) at the Polytechnic University of Catalonia. He has important scientific contributions in the areas of fault diagnosis and fault tolerant control using interval models. He participated in more than 20 international and national research projects in the last decade. He led many private contracts, and published more than 80 articles in JCR journals and more than 350 in international conference/workshop proceedings.

Prof. Puig supervised over 20 PhD theses and over 50 MA/BA theses.