Data-driven Leak Localization in Water Distribution Networks via Dictionary Learning and Graph-based Interpolation

Paul Irofti[1], Luis Romero-Ben[2], Florin Stoican[1,3], Vicenç Puig[2]

Abstract— **In this paper, we propose a data-driven leak localization method for water distribution networks (WDNs) which combines two complementary approaches: graph-based interpolation and dictionary classification. The former estimates the complete WDN hydraulic state (i.e., hydraulic heads) from real measurements at certain nodes and the network graph. Then, we append to the actual measurements a subset of relevant estimated states to feed and train the dictionary learning scheme. Thus, the meshing of these two methods is explored, and several promising performance results are attained, even deriving different mechanisms to increase the resilience to classical issues (e.g., dimensionality, interpolation errors, etc.). The approach is validated using the L-TOWN benchmark proposed in the BattLeDIM2020 competition.**

Index Terms— **leak localization, dictionary learning, interpolation, water distribution networks.**

I. Introduction

Fault detection and isolation (FDI) is an unavoidable element in any engineering system that is operated in an autonomous manner. This has become especially relevant in the last decades due to the continued increase in complexity (in terms of size and interconnections) of networked systems. Water distribution networks (WDNs) are a prime example: they involve hundreds or even tens of thousands of nodes, have poor observability (traditionally outflows are measured only in tanks and reservoirs) and conservative control architecture (i.e., communication and decisions flow rigidly). All these issues make fault events (pipe bursts leading to leakage) hard to detect and isolate accurately.

Present day WDNs are becoming smarter by the addition of sensors, which provide timely information about pressures (node heads) and debits (pipe flows). This has made possible and raised considerable interest in two questions:

- how should the limited number of sensors be placed in the WDN such as to increase observability required for leak localization [1], [2];
- based on measurements, what is the most efficient method to ensure leak detection and isolation [3], [4].

Paul Irofti and Florin Stoican were supported by grants of the Romanian Ministry of Education and Research, CNCS - UEFISCDI, project number PN-III-P1-1.1-PD-2019-0825 and project number PN-III-P2-2.1-PED-2019-3248, within PNCDI III. Luis Romero and Vicenç Puig want to thank the Spanish national project L-BEST (Ref. PID2020-115905RB-C21), as well as the Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI (MDM-2016-0656).

[1] Research Center for Logic, Optimization and Security (LOS), Department of Computer Science, Faculty of Mathematics and Computer Science, University of Bucharest, Romania, paul@irofti.net

[2] Institut de Robòtica i Informàtica Industrial, Universitat Politècnica de Catalunya, Barcelona, Spain {luis.romero.ben,vicenc.puig}@upc.edu

[3] Politehnica University of Bucharest, Dept. of Automation Control and Systems Engineering, Romania, florin.stoican@upb.ro

Mainly, the difficulty of these questions comes from the problem size. Thus, there is a recent trend in the state of the art to apply data-driven methods [5]. This avoids the need of having an accurate mathematical model required by model-based approaches but neglects some physical relations that exist in the real system. A possible way to compensate the disregarding of these physical relations is by exploiting the system structure, i.e., the network's graph.

Contributions. The above led us to the idea of combining two recently developed and promising approaches: the graph-state interpolation (GSI) procedure proposed in [6] and the dictionary learning (DL) detailed in [7]. In this way, interpolated data (*virtual sensors*), generated in an unsupervised way, is added to the real measurements to feed the learning step of DL, hence indirectly considering structural information during the leak localization. Moreover, the use of a robust classification step [8], provided by DL, allows to improve the node-level classification of GSI.

Thus, this article constitutes an initial step in the development of a method which combines two separate methods such that it compensates their weaknesses and boosts their strengths. Additionally, a new application scheme is proposed to alleviate the drawbacks of introducing approximated data to the learning step of DL: separate dictionaries considering individual extra *virtual sensors* (subsequently combined using a voting rule) are derived, instead of computing a single dictionary that encompasses all those *virtual sensors*.

The theoretical elements are tested over the L-Town benchmark proposed in [9], carrying out simulations on EPANET [10] to generate the required fault scenarios. Initial results show that the methodology is promising in terms of its comparison with the individual methods, settling an appropriate path for future developments.

II. Preliminaries

We briefly introduce the theoretical background used in the rest of the paper.

A. Dictionary Learning Methods

Recently, we successfully applied dictionary learning techniques to solve WDN fault detection and isolation problems [7], [11]. The DL problem trains an over-complete base $\boldsymbol{D} \in \mathbb{R}^{m\times n}$, also called a dictionary, on a dataset $\boldsymbol{Y} \in \mathbb{R}^{m\times N}$ to produce the s-sparse representations $\boldsymbol{X} \in \mathbb{R}^{n\times N}$:

$$\begin{aligned} \min_{\boldsymbol{D},\boldsymbol{X}} \quad & \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{x}_\ell\|_0 \le s,\ \ell = 1:N \\ & \|\boldsymbol{d}_j\| = 1,\ j = 1:n \end{aligned} \tag{1}$$

The s nonzero entries of a given dataset sample's representation define the dictionary columns, also called atoms, and their associated coefficients. A thorough description of the field is given in [12].

For water networks, the dataset consists of sensor node pressure measurements where each sample corresponds to a leak of a given magnitude that occurs in one of the network nodes. The FDI task consists of finding which node contains the leak. Looking at this as a classification problem, where the network nodes represent the classes, we can view the dataset as multiple column blocks corresponding to pressure measurements of leak scenarios occurring in a given node at various magnitudes.

For DL classification tasks, Label Consistent K-SVD (LC-KSVD) [13] extends (1) to simultaneously learn the linear classifier $\boldsymbol{W}$ based on labels $\boldsymbol{H}$ with an added discriminate constraint given by $\boldsymbol{Q}$ that enforces sets of atoms to only be used by a given class

$$\min_{\boldsymbol{D},\boldsymbol{W},\boldsymbol{A},\boldsymbol{X}} \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F^2+\alpha\|\boldsymbol{H}-\boldsymbol{W}\boldsymbol{X}\|_F^2+\beta\|\boldsymbol{Q}-\boldsymbol{A}\boldsymbol{X}\|_F^2 \tag{2}$$

Locating the leaky node from sensor measurements $\boldsymbol{y}$ is a two step process: first we obtain the sparse representations $\boldsymbol{y} \approx \boldsymbol{D}\boldsymbol{x}$ by using a greedy algorithm such as OMP [14]; secondly, we perform linear classification $i = \arg\max \boldsymbol{W}\boldsymbol{x}$ to obtain the leaky node[1] i. To accommodate large sets of data, this approach was extended to the online semi-supervised setting in [15], adapted and applied to large WDNs in [7]. In terms of complexity, performing OMP for one data-item costs us $O(smn)$ operations and updating the dictionary inside LC-KSVD $O(sm^2N)$. Training one dictionary involves K OMP and dictionary update iterations.

B. Interpolation strategies

In the past, we have used interpolation methods as part of leak localization schemes [6]. In particular, this interpolation process works over hydraulic head (node pressure + elevation) data, measured by pressure sensors (cheaper and easier to install than other metering systems [16]).

The network structure is considered to be represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, composed by a set of nodes, i.e., $\mathcal{V} = \mathcal{R} \cup \mathcal{J}$, that model the reservoirs ($\mathcal{R}$) and junctions ($\mathcal{J}$) of the WDN; and a set of edges, i.e., $\mathcal{E}$, representing the pipes of the network. The proposed interpolation approximates the actual relation, given by the Hazen-Williams formula, between the junctions' hydraulic heads using the following linear relation:

$$f_i = \frac{1}{\phi_i}\boldsymbol{\omega}_i \boldsymbol{f} \tag{3}$$

where $\boldsymbol{f} \in \mathbb{R}^{|\mathcal{V}|}$ is the complete graph-state vector, which encompasses the estimated and known hydraulic head values; $\boldsymbol{\omega}_i$ stands for the *i-th* row of the weighted adjacency matrix $\boldsymbol{\Omega} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, which encodes the connectivity among nodes as well as the strength of these connections; and $\phi_i = \sum_{j=1}^{|\mathcal{V}|} \omega_{ij}$ denotes the *i-th* element of the diagonal of the degree matrix $\boldsymbol{\Phi} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, which is a diagonal matrix. The weights in the matrix $\boldsymbol{\Omega}$ are derived from the actual pipe lengths: considering p_{ij} to be the length of the pipe connecting nodes $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ ($p_{ij} = 0$ if they are not adjacent), then $\omega_{ij} = \frac{1}{p_{ij}}$ if $p_{ij} \neq 0$, and $\omega_{ij} = 0$ otherwise. This selection increases the effect of closer neighbors over the state of a node.

Considering the previous approximation, the interpolation procedure can be expressed as a quadratic programming problem (see [6] for the complete development):

$$\min_{\boldsymbol{f}} \quad \frac{1}{2}\left[\boldsymbol{f}^T\boldsymbol{L}\boldsymbol{\Phi}^{-2}\boldsymbol{L}\boldsymbol{f} + \alpha\gamma^2\right] \tag{4a}$$
$$\text{s.t.} \quad \boldsymbol{B}\boldsymbol{f} \leq \boldsymbol{1}_{|\mathcal{V}|}\cdot\gamma \tag{4b}$$
$$\gamma > 0 \tag{4c}$$
$$\boldsymbol{S}\boldsymbol{f} = \boldsymbol{f}^s \tag{4d}$$

where $\boldsymbol{L}$ stands for the graph Laplacian, $\boldsymbol{B}$ is the incidence matrix, $\boldsymbol{S}$ is a diagonal matrix with 1 at the elements whose associated node is sensorized (and 0 otherwise), and $\boldsymbol{f}^s$ denotes the measurements vector, including the known hydraulic heads at the elements corresponding to metered nodes (and 0 elsewhere). α and γ are auxiliary scalars which control the relative importance of the cost terms in (4a) and, respectively, relax the direction inequality constraint (4b).

The incidence matrix $\boldsymbol{B}$ deserves some additional explanation. Structural information (node position and elevation, pipe length) is usually available but information about flow direction within the pipes is much harder to get, due to the rare availability of flow meters. A possible way to overcome this issue is to compute an approximated incidence matrix only using the network structure: the shortest path from each reservoir to each junction node is calculated, storing the path, i.e., list of ordered nodes, for every possible combination. Considering that each pipe is defined by two nodes, we can count, for every stored path, how many times a pipe is traversed following each one of the two possible directions, adding all the counts regarding a certain pipe and a certain direction to obtain its final count (note that, at the end of the process, each pipe would have two assigned final counts, one per possible direction). Then, the flow direction along a pipe is taken as the direction with the higher final count.

In terms of complexity, the interpolation procedure reduces to solving quadratic problems (4) whose complexity is polynomial in terms of problem dimension (e.g., solving N problems with the interior point method gives $O(n^{3.5}N)$).

III. Methodology

The methodologies presented in Section II have been successfully applied to solve the leak localization task, as illustrated in the provided references. Nevertheless, the performance of these techniques is limited by different aspects:

- The DL approach introduced in Section II-A provides a satisfactory node-level localization, i.e., the leaks are mostly correctly located at the node where they appear. However, the method does not exploit the structure of

[1] Note that while $\boldsymbol{Y}$ contains only sensor node information, the labels in $\boldsymbol{H}$ help to identify leaky non-sensorized nodes.

the graph during the learning process, missing important information that may improve the performance.
- The GSI technique presented in Section II-B, together with the localization strategy proposed in [6], explicitly uses the graph structure for the sake of the leak isolation. However, the node-level localization precision is lower, as the method was originally conceived to isolate an area where the leak is located.

Thus, we propose a combination of GSI and DL that maximizes their benefits, leading to the presented leak localization methodology, henceforth referred as GSI-DL (note that the leak detection phase is not pursued here and is assumed to be realized through one of the techniques from the state of the art, e.g, by tracking changes in the night consumption [17]). Then, GSI-DL consists of the next steps:

1) The hydraulic dataset (hydraulic heads at the sensorized nodes) is generated or provided, considering different leak scenarios (leak location, magnitude, occurring time, etc.), as well as leak-free historical data.
2) The complete hydraulic state of the network (represented by the hydraulic heads at the WDN nodes) is estimated from the measured pressure values by means of GSI, for both the leak and leak-free scenarios, from which we derive the complete residuals dataset.
3) A subset of nodes is selected to play the role of *virtual sensors*, i.e., their interpolated state value is added to the information provided by the real sensors, obtaining an assembled residuals dataset.
4) The DL algorithm is fed with the assembled residuals dataset, performing the training and obtaining the corresponding dictionary and linear classifier.

To conclude, the aim of the combined GSI-DL method is to merge the complementary strengths of each individual component (i.e., GSI supplies additional information for DL) in order to improve the classification performance.

A. Selection of the virtual nodes

The interpolation scheme retrieves the complete WDN state from a reduced set of known values from the physical sensors subset $\mathcal{S}_r \subset \mathcal{V}$. However, the introduced approximations cause differences between the actual hydraulic heads and the computed states at the unknown nodes, with an error distribution that strongly depends on the number and placement of sensors because they are the unique source of hydraulic data.

Considering the importance of the existence of distinctive features for each leak, the insertion of estimated data must be carefully considered to avoid the inclusion of nodes whose values present high differences between actual and estimated value, between leaky and nominal data, etc., which can hinder the DL process and reduce the localization accuracy.

Thus, a grid search is performed over the set of possible *virtual sensors* subsets $\mathcal{S}_v$ via training and testing to see if accuracy improves significantly on the training data. At the end, we select the best *virtual sensors* in terms of accuracy. In this way, the selected *virtual sensors* are very likely to add valuable information to the learning process and to increase separability among leaks.

B. GSI-DL procedure

To perform dictionary learning, the first step consists on collecting readings from pressure sensors in a wide variety of scenarios (as training samples are required): nominal/leaky, as well as different leak sizes and locations. This task may be solved in two different ways:

- On the one hand, synthetic sensor data can be generated from a known hydraulic model. For a better performance assessment, we use this approach for this article.
- On the other hand, if the network is clustered to define groups of nodes whose leak behaviour is similar, a set of leak experiments can be performed together with the water utility at the real WDN to gather the sensor data.

In our case, considering the typically large size of real networks, as well as the computational cost of performing simulations for every possible leak event, a subset $\mathcal{Z} \subseteq \mathcal{V}$ of nodes is chosen so that $|\mathcal{Z}| = c \leq N$ leak sources (labels for training) are considered, running scenarios with various fault magnitudes for each node $\varkappa_i$. This subset of nodes must be scattered throughout the network to capture its hydraulic behaviour in the most complete way.

Noting that M different leak sizes are considered for each possible leak location (t time instants are computed considering each leak size), and that the number of physical sensors of the network is $|\mathcal{S}_r|$, the complete data set $\tilde{\boldsymbol{F}}_{leak} \in \mathbb{R}^{|\mathcal{S}_r| \times \psi}$, with $\psi = cMt$, may be regarded as the union of c blocks corresponding to each faulty node $\varkappa_i$, where each block contains Mt samples with M different leak sizes.

Moreover, a complementary leak-free/nominal dataset $\tilde{\boldsymbol{F}}_{nom} \in \mathbb{R}^{|\mathcal{S}_r| \times \psi}$ is required: each column $\tilde{\boldsymbol{f}}_{nom}$ must be obtained for similar boundary conditions at the WDN to the case of its analogue (by position) column $\tilde{\boldsymbol{f}}_{leak}$ from $\tilde{\boldsymbol{F}}_{leak}$. In this way, the necessary residuals for DL are obtained while simultaneously reducing the differences between leak and leak-free scenarios that are not caused by the leak.

The achieved datasets must be divided into training and testing sets. Then, the learning process, explained in Section II-A, is applied as summarized by Algorithm 1 (consider that $s_{rv} = |\mathcal{S}_r| + |\mathcal{S}_v|$ is the total number of sensors).

The obtained matrices, i.e., $\boldsymbol{D}$ and $\boldsymbol{W}$, are then used to classify the entries of the testing dataset in order to assess the reliability of the solution. Algorithm 2 summarizes the behavior of the classifier against new data entries.

C. GSI-DL with multiple dictionaries

The application of the strategy is limited due to the insertion of data from multiple interpolated nodes, which leads to an accumulation of approximation errors and, above a threshold, deteriorates the leak localization efficacy. Thus, we devise a new scheme to overcome the above where, instead of learning a single dictionary-classifier pair that is trained with the entire set of additional *virtual sensor* information, we learn multiple dictionaries (MD)-classifiers pairs by applying Algorithm 1 separately for each individual *virtual sensor*.

Algorithm 1: GSI-DL — Training procedure

Data: leak training set $\tilde{\boldsymbol{F}}^{train}_{leak} \in \mathbb{R}^{|\mathcal{S}_r| \times \psi_{tr}}$, nominal training set $\tilde{\boldsymbol{F}}^{train}_{nom} \in \mathbb{R}^{|\mathcal{S}_r| \times \psi_{tr}}$, sparsity level $s \in \mathbb{R}$, *virtual sensors* $\mathcal{S}_v$, physical sensors $\mathcal{S}_r$

Result: dictionary $\boldsymbol{D} \in \mathbb{R}^{s_{rv} \times n}$, classifier $\boldsymbol{W} \in \mathbb{R}^{c \times n}$

interpolate $\boldsymbol{F}^{train}_{nom}$ applying (4) to $\tilde{\boldsymbol{F}}^{train}_{nom}$

interpolate $\boldsymbol{F}^{train}_{leak}$ applying (4) to $\tilde{\boldsymbol{F}}^{train}_{leak}$

compute residuals: $\boldsymbol{Y}^{train} = \boldsymbol{F}^{train}_{nom} - \boldsymbol{F}^{train}_{leak}$

assemble dataset (and associated labels $\boldsymbol{H}^{train}_{rv}$): $\boldsymbol{Y}^{train}_{rv} = \left[\boldsymbol{Y}^{train}(\mathcal{S}_r); \boldsymbol{Y}^{train}(\mathcal{S}_v)\right]$

compute $\boldsymbol{D}, \boldsymbol{W}, \boldsymbol{A}$ applying (2) to labeled $\boldsymbol{Y}^{train}_{rv}$

Algorithm 2: GSI-DL — Classification procedure

Data: sample $\tilde{\boldsymbol{f}}^{test}_{leak} \in \mathbb{R}^{|\mathcal{S}_r|}$, nominal sample $\tilde{\boldsymbol{f}}^{test}_{nom} \in \mathbb{R}^{|\mathcal{S}_r|}$, sparsity level $s \in \mathbb{R}$, *virtual sensors* $\mathcal{S}_v$, physical sensors $\mathcal{S}_r$, dictionary $\boldsymbol{D} \in \mathbb{R}^{s_{rv} \times n}$, classifier $\boldsymbol{W} \in \mathbb{R}^{c \times n}$

Result: faulty node i

interpolate $\boldsymbol{f}^{test}_{nom}$ applying (4) to $\tilde{\boldsymbol{f}}^{test}_{nom}$

interpolate $\boldsymbol{f}^{test}_{leak}$ applying (4) to $\tilde{\boldsymbol{f}}^{test}_{leak}$

compute residuals sample: $\boldsymbol{y}^{test} = \boldsymbol{f}^{test}_{nom} - \boldsymbol{f}^{test}_{leak}$

assemble data: $\boldsymbol{y}^{test}_{rv} = \left[\boldsymbol{y}^{test}(\mathcal{S}_r); \boldsymbol{y}^{test}(\mathcal{S}_v)\right]$

representation: $\boldsymbol{x} = \text{OMP}(\boldsymbol{y}^{test}_{rv}, \boldsymbol{D}, s)$

classification: $i = \arg\max_{j=1:c}(\boldsymbol{W}\boldsymbol{x})$

The result consists of $|\mathcal{S}_v|$ dictionaries-classifiers pairs, each one trained with a single additional *virtual sensor*.

For the final localization decision, we employ a voting method [18] on the set of classifications from the dictionary-classifier pairs. Several advantages are derived from the application of this scheme, chiefly, a more robust classification result [8] and a reduction in the outliers' effect.

IV. Results

The presented methodologies are implemented in a realistic case study, provided by the "Battle of Leakage Detection and Isolation Methods 2020" (BattLeDIM2020), detailed in [9]. This benchmark, illustrated in Fig.1, consists of a small hypothetical network composed of 782 inner nodes, 909 pipes, 1 tank and 2 water inlets or reservoirs. The network is composed from three distinct areas (A, B and C), distinguished by the elevation of its nodes. Hereinafter, we focus on area A.

A. Data generation

For testing purposes, hydraulic data must be available for all the considered leaks. In this article, this information is retrieved through the simulation of an EPANET model.

A list $\mathcal{Z} \subset \mathcal{V}$ of 30 nodes is selected from the junction nodes (denoted in Fig.1 with "filled blue circle" symbols) as possible leak sources ("red cross" symbols). Henceforth, 30

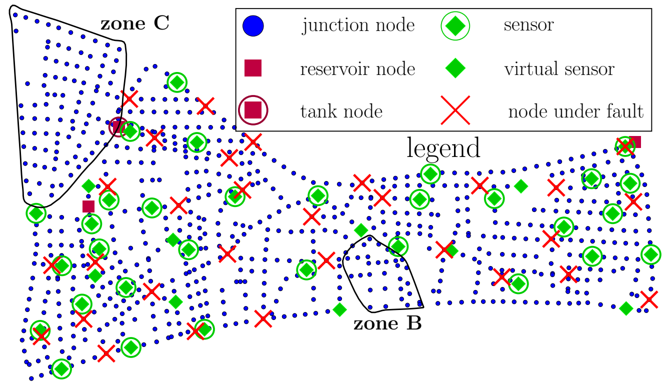

Fig. 1. L-Town illustration with various node highlights

labels (one per each leak) are considered in the subsequent learning steps. For each possible leak, a week of hydraulic data is generated, with a time step of 5 minutes. The first four days of the week are selected to generate the training set, increasing the leak size each day to cover a wide range of values, from 1 m^3/h to 7 m^3/h. The three remaining days produce the testing set, selecting even leak sizes between the minimum and maximum values to complete the set, i.e., 2, 4 and 6 m^3/h. Finally, a 5% of uncertainty is induced in the pipe roughness and diameters, whereas the demand patterns are kept as provided by the BattLeDIM2020 case.

B. Application of preliminaries

First, the preliminary works in Section II, i.e., the separated application of DL [7] and GSI (together with a leak candidate selection method: GSI-LCSM) [6]; are applied to the case study. This is done to provide performance baselines and to justify the interest in merging the procedures.

On the one hand, the classical DL approach is trained with the measurements from the real sensors, obtaining an accuracy of 89.63%, which can be considered as the reference value to improve with the new proposed schemes that include the interpolated information. This accuracy is included in Table I together with the rest of the single-dictionary results, as they are directly comparable.

On the other hand, GSI-LCSM is fully data-driven (it does not require prior leak information) and is conceived to provide a candidate localization area, which limits the precision for single node localization. Thus, we need to select a suitable criteria to determine the success of the localization process, in order to perform a comparable analysis with respect to the DL ones. Let us consider that GSI-LCSM provides node v_G as the best candidate. If we only accept localization results as successful when the distance from v_G to v_l (node of the actual leak), i.e, $r(v_G, v_l)$; is the minimum among the distances from v_G to all the possible leaks, a reduced accuracy of 56.7% is achieved. However, this criteria can be extended to also consider the case when v_l is not the closest to v_G, but the 2$^{\text{nd}}$-nearest one (let us denote the closest one as v_c). Accepting only the concrete cases when $r(v_G, v_l) - r(v_G, v_c) < 100$ (meters), the accuracy is drastically increased, reaching a 76.7%. Finally, an accuracy of 86.7% can be achieved if we accept as successful all the cases when v_l is at least the 2nd-nearest possible leak to v_G.

Hence, it is concluded that the GSI-LCSM method successfully reduces the uncertainty in the leak's position to a small zone around it, but the performance deteriorates if analysing to the node-level. This justifies the interest in combining GSI and DL, demonstrated to be suitable for node-level localization tasks.

C. Single dictionary with virtual sensors

We apply the method in Section III-B to select a set of *virtual nodes*, as per the criteria explained in Section III-A. Thus, GSI is used over a leak dataset having $|\mathcal{S}_r| = 33$ measurements (from 29 nodes with pressure sensors, 2 reservoirs' pressure sensors, and other 2 known pressures at the PRVs; depicted with a "green circled diamond" in Fig. 1), $c = 30$ different leaks, $M = 7$ different leak sizes and $t = 288$ time slots (at every 5 minutes in 24 hours).

The classification accuracy for a set of considered *virtual sensors* is included in Table I, whereas their location within the network is shown in Fig. 1 ("green diamond" markers). For each table entry, a different subset of *virtual sensors* is selected from the GSI interpolation. Each accuracy value is averaged over five training runs, to reduce the effect of possible learning outliers.

TABLE I

SINGLE DICTIONARY - ACCURACY RESULTS

Virtual node/s	**Testing accuracy (%)**
None	89.63
255	90.54
559	90.23
51	90.43
153	90.09
504	90.57
339	91.96
265	89.77
773	82.53
214	81.34
265, 255	89.42
265, 255, 559	88.54

Comparing the first entry (the case without *virtual sensors*) against the rest, we conclude that:

- generally, the addition of a single *virtual sensor* (from node 255 to node 265 in the table) increases the accuracy of the classification, and thus, the leak localization;
- there exist nodes whose interpolated information degrades the performance (due to discrepancies between their nominal and leaky interpolated data), as seen, e.g., for nodes 773 and 214 in the table;
- lastly, the insertion of more than one interpolated value gradually lowers the accuracy, as shown in the last two entries of the table.

Worth mentioning is also the associated computational complexity. Increasing the number of sensors by adding virtual ones leads to larger computation times during DL. We show in Table II a time consumption analysis, carried for the same DL training parameters.

TABLE II

SINGLE DICTIONARY - TIME CONSUMPTION

No. of added *virtual sensors*	**Time consumption (s)**
0	97
193	119
448	150
628 (zone A + reservoirs)	188

D. Multiple dictionaries and voting methods

The performance degradation observed in Section IV-C, associated to the inclusion of multiple *virtual sensors*, justifies the development of the scheme introduced in Section III-C. Thus, multiple dictionary-linear classifier pairs are derived, considering a single *virtual sensor* for each case. Besides, a voting method is applied to improve the classification result. In this case, a "plurality voting" scheme has been selected, i.e., the label with the maximum number of votes among the different dictionary-classifier pairs is selected as the definitive label for the classified sample. We apply the voting scheme to two different implementations, and present performance results in Table III, including the classification accuracy (Acc.) and the model training time (Time):

- (MD)DL: the classical DL construction is applied, with consecutive increase for the number of dictionaries;
- (MD)GSI-DL: each dictionary adds a single *virtual sensor* (distinct from all the others inserted a priori), again, the number of dictionaries is augmented consecutively.

TABLE III

MULTIPLE DICTIONARIES - ACCURACY RESULTS

No. of dicts.^{*1}	**(MD)DL**		**(MD)GSI-DL**	
	Acc. (%)	Time (s)	Acc. (%)	Time (s)
1^{*2}	94.40	247	94.40	272
2	93.91	494	93.57	525
3	95.50	740	95.38	773
4	95.17	985	95.78	1017
5	96.09	1231	96.54	1257
6	95.78	1472	96.36	1499
7	96.09	1724	96.54	1741

^{*1}Considered *virtual sensors*: 255, 559, 51, 153, 504, 339.
^{*2}The first dictionary is a classical DL for both cases.

E. Discussion

Several conclusions may be derived from the results in Table III, regarding the suitability of the proposed method.

Interestingly, the classification performance for both methods is almost identical for the first three entries (to be expected for the first entry since here a classical dictionary is computed in both cases) with significant improvements appearing only when considering four or more dictionaries in the voting step. This shows that subsequent dictionary-classifier pairs complement the classification performance of the standard DL, strengthening the common correct results and providing extra information to overcome possible weaknesses. Moreover, we note that (MD)GSI-DL surpasses (MD)DL when extra dictionary-classifier pairs with single *virtual sensors* are included, confirming the appropriateness of this methodology over the single-dictionary approach.

Besides, a comparison among the original and proposed approaches presented in this article, i.e., DL (with a single dictionary), GSI-LCSM, (MD)DL and (MD)GSI-DL, is represented in Fig. 2. The first two methods are used as baselines, and hence they are included in Fig. 2 as dashed lines, despite only one or even no dictionary is used in their resolution. Then, for the proposed multiple dictionary (MD) approaches, the performance evolution regarding the number of "different" dictionaries (three dictionaries of the same kind are used at each entry of Table III) is included.

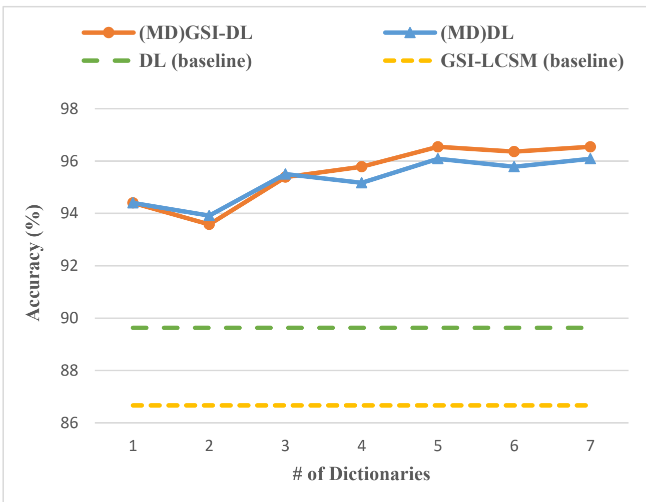

Fig. 2. Performance comparison among the original and proposed methods

This comparison illustrates the advantages of the MD scheme, which increases the accuracy from 89.63% to 94.4% using the same base method (classical DL) with the only difference of deriving extra dictionaries and voting among them. Moreover, the improvement of (MD)GSI-DL with respect to (MD)DL is graphically represented to highlight the benefits of including *virtual sensors* at the extra dictionaries.

In conclusion, these results show the promising characteristics of the proposed methodology, justifying the continuation on this research line to develop future improvements that yield a rounded data-driven leak localization strategy.

V. Conclusions

This article proposes a method for leak localization in WDNs that combines graph-based state interpolation (GSI) and dictionary learning (DL). We have presented the original methods, explaining advantages and drawbacks to show and justify the interest in their complementary application.

Furthermore, several different schemes have been introduced regarding this combination, i.e., computing a single dictionary that considers a set of interpolated values, deriving multiple dictionaries (classical or including *virtual sensors*) and voting over their classification results, etc.

The advantages and limitations of all these approaches have been stated and demonstrated by means of a case study, based on the L-Town benchmark from BattLeDIM2020, obtaining several promising results.

Further works will cover additional enhancements of the methodology, such as a better integration of the components, an analysis of the interpolation procedure to improve its performance (weight computation, nonlinear fitting, etc.).

References

[1] A. Deshpande, S.E. Sarma, K. Youcef-Toumi, and S. Mekid, "Optimal coverage of an infrastructure network using sensors with distance-decaying sensing quality," *Automatica*, vol. 49, no. 11, pp. 3351–3358, 2013.

[2] L.S. Perelman, W. Abbas, X. Koutsoukos, and S. Amin, "Sensor placement for fault location identification in water networks: A minimum test cover approach," *Automatica*, vol. 72, pp. 166–176, 2016.

[3] 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.

[4] J. Meseguer, J.M. Mirats-Tur, G. Cembrano, V. Puig, J. Quevedo, R. Pérez, G. Sanz, and D. Ibarra, "A decision support system for on-line leakage localization," *Environmental modelling & software*, vol. 60, pp. 331–345, 2014.

[5] A.-W. Taha, S. Sharma, and M. Kennedy, "Methods of assessment of water losses in water supply systems: a review," *Water Resources Management*, vol. 30, no. 14, pp. 4985–5001, 2016.

[6] 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.

[7] 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.

[8] Michael R Smith and Tony Martinez, "The robustness of majority voting compared to filtering misclassified instances in supervised classification tasks," *Artificial Intelligence Review*, vol. 49, no. 1, pp. 105–130, 2018.

[9] S. G. Vrachimis, D. G. Eliades, R. Taormina, A. Ostfeld, Z. Kapelan, S. Liu, M. Kyriakou, P. Pavlou, M. Qiu, and M. M. Polycarpou, "BattLeDIM: Battle of the Leakage Detection and Isolation Methods," 2020.

[10] L. A. Rossman, *EPANET 2 User's Manual*, Washington, D.C., 2000.

[11] 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.

[12] B. Dumitrescu and P. Irofti, *Dictionary Learning Algorithms and Applications*, Springer, 2018.

[13] 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, Nov. 2013.

[14] Y.C. Pati, R. Rezaiifar, 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.

[15] P. Irofti and A. Băltoiu, "Malware identification with dictionary learning," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[16] 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.

[17] R. Puust, Z. Kapelan, D.A. Savić, and T. Koppel, "A review of methods for leakage management in pipe networks," *Urban Water J.*, vol. 7, no. 1, pp. 25–45, 2010, https://doi.org/10.1080/15730621003610878.

[18] M. Van Erp, L. Vuurpijl, and L. Schomaker, "An overview and comparison of voting methods for pattern recognition," in *Proceedings 8th International Workshop on Frontiers in Handwriting Recognition*. IEEE, 2002, pp. 195–200.