GraphSmart: A Method for Green and Accurate IoT Water Monitoring

TIZIANA CATTAI, Department of Information Engineering, Electronics and Telecommunications, Sapienza University of Rome, Rome, Italy
STEFANIA COLONNESE, Departement of Information Engineering, Electronics and Telecommunications, Sapienza University of Rome, Rome, Italy
DOMENICO GARLISI, Computer Science Department, University of Palermo, Palermo, Italy and CNIT, Parma, Italy
ANTONINO PAGANO, Department of Engineering, University of Palermo, Palermo, Italy and CNIT, Parma, Italy
FRANCESCA CUOMO, Departement of Infomation Engineering, Electronics and Telecommunications, Sapienza University of Rome, Rome, Italy

Water scarcity is nowadays a critical global concern and an efficient management of water resources is paramount. This paper presents an original approach for monitoring Water Distribution Systems (WDSs) through Internet of Things (IoT) that involves the integration of multiple sensors placed across the distribution network to accurately measure water flow. To enhance energy efficiency for green monitoring and communication process, we harness the power of graph theory and graph signal processing to represent in a tunable and accurate way the water flow and simultaneously minimize the number of IoT sensors communicating those measurements. We propose a graph model where water flow is represented as signal on graph and we introduce an algorithm, named GraphSmart, designed to reconstruct the graph signal when certain measurements are unknown or missing. Our framework is applied on a synthetic realistic environment within the context of LoRaWAN (Long Range Wide Area Network), an infrastructure and protocol designed for ultra-low-power IoT devices. Our findings show that GraphSmart significantly reduces energy consumption while ensuring precise flow estimation. Our research demonstrates high potential for energy-efficient and accurate water flow monitoring, paving the way to improve the management of WDSs and enabling water operators to address water scarcity challenges.

This work was supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART") in the WITS (Watering IoTs) and SPRINT focused projects.
Authors' Contact Information: Tiziana Cattai, Department of Information Engineering, Electronics and Telecommunications, Sapienza University of Rome, Rome, Italy; e-mail: tiziana.cattai@uniroma1.it; Stefania Colonnese, Departement of Information Engineering, Electronics and Telecommunications, Sapienza University of Rome, Rome, Lazio, Italy; e-mail: stefania.colonnese@uniroma1.it; Domenico Garlisi, Computer Science Department, University of Palermo, Palermo, Italy and CNIT, Parma, Italy; e-mail: domenico.garlisi@unipa.it; Antonino Pagano, Department of Engineering, University of Palermo, Palermo, Italy and CNIT, Parma, Italy; e-mail: antonino.pagano@unipa.it; Francesca Cuomo, Departement of Infomation Engineering, Electronics and Telecommunications, Sapienza University of Rome, Rome, Lazio, Italy; e-mail: francesca.cuomo@uniroma1.it.

ACM 1550-4859/2024/11-ART130
https://doi.org/10.1145/3695769

CCS Concepts: • **Networks** → **Sensor networks;**

Additional Key Words and Phrases: Graph network, smart water distribution networks, IoT, flow reconstruction, wireless sensors, LoRaWAN, energy efficiency

ACM Reference Format:
Tiziana Cattai, Stefania Colonnese, Domenico Garlisi, Antonino Pagano, and Francesca Cuomo. 2024. GraphSmart: A Method for Green and Accurate IoT Water Monitoring. *ACM Trans. Sensor Netw.* 20, 6, Article 130 (November 2024), 32 pages. https://doi.org/10.1145/3695769

1 Introduction

Water scarcity is an ever-growing concern that affects communities and ecosystems worldwide [1, 2]. The increasing demand for clean, potable water, together with the impact of climate change, has motivated the research on improving the use of water resources. According to government reports and available data [3], the result provided by the European Commission for the Europe region is approximately 26% of wasted water, with a spike of 45% in some specific areas [4].

The scarcity of water underscores the need for particular attention to be paid to the correct and efficient use of **Water Distribution Systems (WDSs)** also given their poor condition due to aging and unmaintained infrastructures.
This motivates the design of new technological and modeling tools for effective monitoring. A WDS comprises water sources, water treatments, pumping stations, a **water distribution network (WDN)**, and finally, the end users. A WDN encompasses junctions, tanks and reservoirs pipes, pumps, and valves. The junctions can serve as water supply or demand points to and from the network, while the tank(s) or reservoir(s) inject water into the system and represent finite or virtually infinite water sources.
Traditional monitoring approaches have often relied on manual inspections or simplistic models, which can be time-consuming, labour-intensive, and prone to errors [5].

For these reasons, we focus on the possibility to monitor WDNs with the **Internet of Things (IoT)**, specifically using sensor meters measuring the water flow, that employ low-power wireless technologies to transfer information to central data collection and processing centers. However, two important aspects must be considered:

(1) It is very difficult and costly to equip all branches of the distribution network with water sensors;
(2) Some of these sensors might not function properly leading to inaccurate measurements or due to battery issues may not measure at all the flow giving rise to "unmeasured nodes", as indicated in the following.

Therefore, we propose an approach that aims to optimize the use of the available sensors by identifying which sensors are the best to use. Our target is twofold: (i) to maximize the accuracy of water flow measurements (represented as a signal on a graph) and (ii) to minimize the energy consumption associated with data collection and transmission. By optimizing either the positions of the sensors or selecting which sensors transmit their measured data, we aim to maximize data accuracy while minimizing energy consumption, even in the presence of noise. This approach not only enhances the efficiency and reliability of the water distribution system, but also ensures that the monitoring solution is both cost-effective and scalable. It is possible to include this method inside the more general framework in [6]. As far as we know, an approach that simultaneously considers the quality of the reconstructed measurement and the energy consumption of the IoT sensors for water measures transmission has not been previously addressed in the literature.

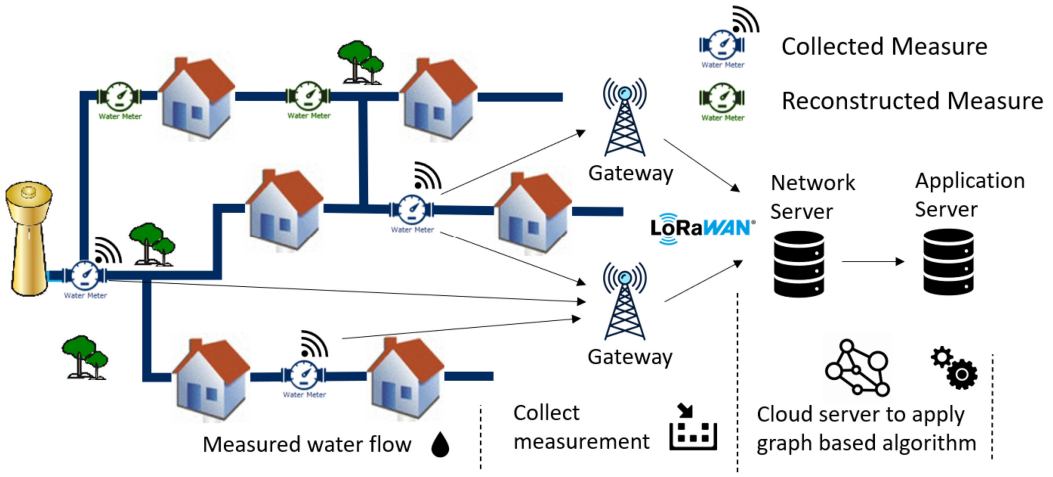


Fig. 1. A LoRaWAN architecture where LoRaWAN devices are positioned along pipes within the WDS to collect data from meter sensors.

The considered scenario is the following (Figure 1): Wireless sensor meters are connected with one or more **Gateways (GWs)** that forward packets to a **Network Server (NS)** and to the corresponding IoT applications servers. In this context, challenges in water monitoring using IoT devices are:

(1) deploying each sensor incurs a cost (for both the device and the deployment itself);
(2) each sensor consumes energy when transmitting its data;
(3) certain areas or pipes may be inaccessible for sensor node placement;
(4) some sensors might be physically damaged or have low battery levels.

Theoretically, we could place a sensor meter (i.e., a measurement point) at every link of a WDN, but given the aforementioned issues, in practice, only a subset of sensors can be used for monitoring.

Concerning the communication technology to be used by sensor devices, there are no universal standards for WDS monitoring. IoT chips with low power consumption and long-distance wireless communication capability are ideal for these purposes, like WiFi-based, cellular networks or **LoW Power WAN (LPWAN)**. LPWAN devices are expected to dominate the field [7], with different infrastructure requirements: (i) cellular infrastructure-dependent, such as NB-IoT; (ii) third-party infrastructure reliant, like SigFox; and (iii) Autonomous LPWANs, such as **Long Range Wide Area Network (LoRaWAN)** [8]. Cellular technology-based LPWANs offer wide coverage, capacity, battery life, quality of service, and security but are not cost-effective due to subscription fees and dependence on commercial networks. SigFox [9], a patented network, spurred rapid innovation by increasing competition among LPWAN technologies. LoRaWAN offers numerous benefits, including low power consumption, extensive coverage, simplicity, and easy management due to its characteristics. However, it faces potential scalability issues in large-scale scenarios. This study targets the LoRaWAN technology, enabling real-time analysis and processing of the vast amounts of data generated by connected devices within the network edge.

We propose GraphSmart, a graph-based method designed to increase the precision of water flow reconstruction and boost energy efficiency within the WDS. Leveraging graph theory, we represent and analyze the topology of a distribution network, followed by a graph transformation that aims to extract the minimum set of nodes needed for accurate network flow reconstruction.

In addition, we propose an extended model to include the possibility that some nodes draw water from the network. We refer to this additional virtual flow as "demand values".

We first apply our method to a use case with the aim to deeply understand the functioning of GraphSmart method in different scenarios, such as with and without demand values.

Then, the proposed approach is applied to LoRaWAN technology, identified as the most suitable IoT solution for the specific scenario. We develop a model that incorporates considerations for sensor energy consumption due to their communication ability, acknowledging the non-uniform energy usage of individual nodes due to variations in wireless coverage and radio parameters. This model accounts for the peculiarity of single-node energy consumption within the context of the wireless environment. In order to perform the simulation of GraphSmart in a realistic setting, we use the **EPANET (US Environmental Protection Agency water NETwork)** software and the NS-3 simulator. These tools are widely recognized in the literature as the most suitable for evaluating WDN and LoRaWAN scenarios, respectively. We have analyzed a practical WDN network and presented its corresponding LoRaWAN integration, considering optimal GWs placement and radio coverage parameters. Furthermore, we provide an in-depth analysis of the energy consumption profile of LoRaWAN devices, which is used to calibrate the simulation parameters. In this scenario, we perform additional analyses to merge considerations as for the accuracy in reconstructing the flow from a subset of measurements and the energetic consumption of the communication network. Our findings prove the potential and the innovation of GraphSmart to obtain a green and accurate IoT water monitoring.

Taken together, the key contributions of this paper are:

- We introduce a graph model for WDSs, based on **Graph Signal Processing (GSP)** theory specifically designed for water flow monitoring. In this manner, it is possible to interpret the water flow as a signal on a graph. We also extend this model to the presence of demand values, namely, the possibility that nodes draw water from the network.
- We provide a method, named GraphSmart, to establish a sensor ranking that considers node centrality measures and we design a green flow reconstruction algorithm. The goal is to reconstruct the water flow for a minimum set of available measure. In addition, the ranking procedures suggest the preferred locations to place those sensors.
- We apply our method to a realistic WDN. We demonstrate that our approach simultaneously ensures an accurate flow monitoring and a reduced IoT energy consumption, that means cost savings and enhanced environmental sustainability.
- We provide a synthetic dataset containing realistic WDN settings derived by using the EPANET and NS-3 simulators. It is a publicly available repository in [10]. It can be adopted by the scientific community to test other approaches in a LoRaWAN network communication scenarios applied to WDSs.

The rest of the paper is organized as follows. Section 2 discusses related work on graph-based methods and LoRaWAN applications for WDSs. In Section 3 we introduce the GraphSmart method, while in Section 4 we apply it on several synthetic networks emulating different scenarios. In Section 5 we introduce a study of the LoRaWAN network, including radio parameters and an energy model. In Section 6 we report results of the proposed method on a realistic WDN. Section 8 concludes the paper. For the sake of readability, we report in Table 1 the list of acronyms.

2 Related Work

The presented approach is based on GSP theory specifically designed for water flow monitoring. The main idea is to interpret the water flow as a signal on graph. In this section, we present

Table 1. List of Acronyms

Acronym	Definition	Acronym	Definition
ADR	Adaptive Data Rate	MILP	Mixed Integer Linear Programming
CNN	Convolutional Neural Networks	MSE	Mean Square Error
DER	Data Extraction Rate	NS	Network Server
DFS	Depth First Search	RSSI	Receiver Signal Strength Indicator
DMA	District Meter Areas	SF	Spreading Factor
GHR	Graph-based Head Reconstruction	SoG	Signal on Graph
GSP	Graph Signal Processing	SPQ	Successive Quadratic Programming
GW	Gateway	SVM	Support Vector Machines
IoT	Internet of Things	WDN	Water Distribution Networks
LoRaWAN	Long Range Wide Area Network	WDS	Water Distribution System
LPWAN	Low Power Wide Area Networks	WNTR	Water Network Tool for Resilience

the state-of-the-art research on graph-based approaches, specifically thought for WDS and recent activities on communication technologies for this specific application.

In general, GSP is an emerging research domain recently proposed to analyze data defined on non-Euclidean domains, such as graphs. According to the GSP framework, the structure of a generic graph, that is constituted by nodes and edges, is integrated by signal, that results associated to each graph node. In this context, the operations defined for classical signal processing, such as filtering, compressing and frequency transform, have been reformulated on the graph domain [11]. More precisely, researchers have recently developed methods to capture essential graph characteristics, such as dynamic correlations [12] or non-linearities in the interacting system, that can be described as the succession of states possibly modeled as a Markov model [13] or to multi-kernel learning approaches [14].

GSP theory has found practical applications in various domains, including biological systems [15] and sensor networks [16], where graphs naturally represent the underlying data structure. In the context of the WDSs, which consist of interconnected pipelines and nodal junctions equipped with advanced sensor technology, the graph representation is an intuitive approach. By modeling WDSs as graphs, the physical connections of the system are represented by graph links, and the junctions are represented by graph nodes [17, 18].

In the context of water distribution networks, graph signal processing has recently proved its capabilities [19] in graph learning, in clustering [20] to identify sub-networks and in sectorization projects [21], with particular attention to assess their resilience [22]. Moreover, a large part of the research activity concerning graph theory has mainly focused on the detection of water leakages [23] also using modeling tools [24, 25] and graph neural networks.

The work referenced in our discussion is contrasted with the approach we have presented in Table 2. This table provides a comprehensive overview of various graph-based models employed in water distribution systems, highlighting the different approaches, features analyzed, and tools utilized in each study. Considering the eight pieces of work, we observed that all of them utilize graph theory. Two studies primarily focus on sectorization, while another two concentrate on dimension reduction and redundancy computation, respectively. Three studies are dedicated to leakage detection, with the remaining one centered on pressure estimation. The present work focuses on flow reconstruction for green IoT nodes positioning. The Line Graph model and Node Ranking were employed to analyze flow, utilizing the WNTR and NS-3 tools.

Although graph-based approaches have been explored in the context of water supply systems, many unsolved questions persist. Indeed, accurate and stable methods for WDS monitoring are still missing and other problems related to the IoT application, such as the number of sensors and their location are needed to reconstruct the water flow in the network. In this work, we deal with

Table 2. Comparison of Graph-based Models Applied in WDN

Reference	Study	Nodes/pipes	Models	Features	Tools
[17]	Sectorization based on graph theory	182/282	DFS algorithm	Pressure	WDNetXL
[18]	Network dimension reduction based on graph theory	46/71	SQP technique	Pressure and Flow	–
[21]	Sectorization based on graph theory	195/270	DFS algorithm	Flow	AutoCAD-based computer system SCADRED
[22]	Compute route redundancy and capacity based on graph theory	2416/2638	K-Shortest Paths Index for DMA-based graph	Demand	-
[25]	Leakage Detection based on CNN-SVM and graph theory	12/17	1D-CNN-SVM and a graph model	Leakage	TensorFlow and Python
[24]	Sectorization and leak detection based on graph theory	150/155	MILP formulations	Flow	–
[19]	Pressure reconstruction based on graph theory	100/-	GHR algorithm	Pressure	EPANET
[23]	Leakage detection based on graph theory	268/-	Vertex classification via pagerank	Pressure	WNTR, NetworkX and PyGSP (Python)
Present work	Flow reconstruction based on graph theory	83/85 and 815/1125	Line Graph model and Node Ranking	Flow	WNTR and NS3

the problem of network flow reconstruction in an optimal way. To the best of our knowledge, there are no existing works that combine IoT and graph-based flow reconstruction in the context of WDS.

In this context, the successful application of the aforementioned techniques relies on two crucial factors: the technology used for data collection from sensors and a thorough analysis of the energy impact of the applied methodology. In this regard, the authors in [7] show a comparative analysis of the available IoT technologies in the WDS field. After analyzing extensive literature, Wi-Fi was chosen as the principal technology, followed by LoRaWAN and cellular IoT. Wi-Fi is the most popular choice for IoT applications, but only because the considered literature includes older research scenarios. Indeed, three factors will make LoRaWAN technology dominant in the future WDS scenario. The first factor is the operating frequency. Higher RF frequencies are blocked by walls, trees, and other obstructions, but lower RF frequencies are less susceptible to these issues. That is why cellular and LoRaWAN have lower operating frequencies compared to Wi-Fi. Moreover, if the device is only a sensor node and requires only periodic transmission, then LoRaWAN is a good choice. Two other crucial factors are power consumption and the range of the device. LoRaWAN and cellular have a similar range of about 10 km, but cellular technology has a greater impact on energy consumption. Motivated by these considerations, we have opted to utilize LoRaWAN as the technology to establish the infrastructure for collecting measurements from WDSs.

Finally, regarding the energy impact, sensors with minimal power consumption need to be paired with high-energy-efficient communication technologies to enable the sensors to have energy autonomy for several years. By combining long-range wireless communication with low power consumption, LoRaWAN technology allows for extended battery life [26]. In conclusion, it is important to highlight the significant impact of the proposed solution, particularly in terms of high-level flow reconstruction. By effectively reconstructing the flow within the WDS, our approach enables the possibility of reducing the number of sensors required and consequently the total consumed energy.

3 The GraphSmart Method for Water Monitoring

In this section, we first give an overview of the proposed approach and afterwards, we delve into the definition of its specific implementation steps. Figure 2 shows the workflow of the GraphSmart method, where: (i) we model the flow data acquisition by a graph representation $\mathcal{G}$ of the physical network; (ii) we introduce an alternative graph representation $\mathcal{G}_L$ where we associate

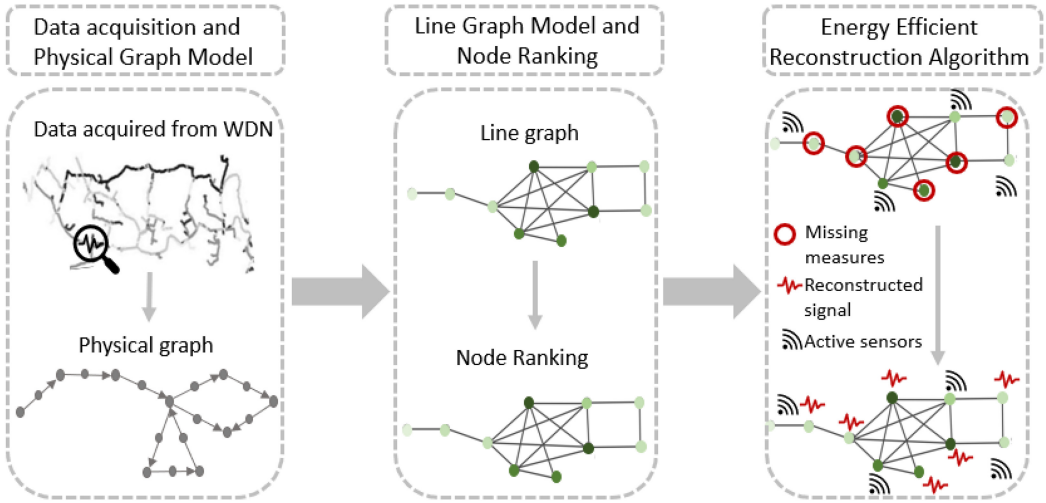


Fig. 2. The figure outlines the proposed method: The first step consists in the generation of a graph related to the physical structure of the network. The second step is the development of an alternative representation through the line graph where flow information is associated to nodes. Then we propose a node ranking by their reconstruction relevance. Finally, we have a reconstruction algorithm that leverages graph topology and selects a minimum set of flow measurements for reconstruction. Specifically, according to the ranking found in the second phase, the measurements of nodes highlighted by red circles (nodes 2,3,5,6,8,10) are not collected, but reconstructed from other nodes (nodes 1,4,7,9) equipped with active sensors.

Table 3. Table of Main Notations

Notation	Description
$\mathcal{G}$,$\mathcal{V}$,$\mathcal{E}$	original graph, set of nodes of $\mathcal{G}$, set of edges of $\mathcal{G}$
$\mathcal{G}_L$,$\mathcal{V}_L$,$\mathcal{E}_L$	line graph, set of nodes of $\mathcal{G}_L$, set of edges of $\mathcal{G}_L$
$\mathbf{A}$, $\mathbf{A}_{FLOW}$, $\mathbf{A}_L$,$\mathbf{A}_{EXT}$	binary adjacency matrix of $\mathcal{G}$, weighted adjacency matrix of $\mathcal{G}$, binary adjacency matrix of $\mathcal{G}_L$, extended binary adjacency matrix with water demands
$\mathbf{B}$,$\mathbf{B}_{FLOW}$	binary incidence matrix, weighted incidence matrix
$\underline{f}$, $\underline{f}^U$,$\underline{f}^K$	flow vector, unmeasured components of $\underline{f}$, measured components of $\underline{f}$

the flow information to the graph nodes, and we rank the nodes based on their relevance for flow reconstruction; and (iii) we provide the reconstruction algorithm based on the knowledge of the graph topology and of a reduced number of flow measurements.

We describe the three steps represented in the figure in the following subsections. For the sake of readability, we report a list of the main notations of the paper in Table 3.

3.1 Data Acquisition and Physical Graph model

The initial step of the algorithm represented in Figure 2 is the mapping of a WDN into a graph to capture the complex structure and dynamics of those systems. According to this representation, the network is translated into a graph $\mathcal{G}$=($\mathcal{V}$, $\mathcal{E}$), that is a mathematical model represented through its N nodes (or vertices) and N_e links (or edges). The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ is here modeled as a binary matrix whose generic element a_{ij} is 1 if there is a pipe between junctions i and j, otherwise it is 0. In the context of water distribution networks, the standard way to build the graph consists in associating links to pipes and nodes to points where the pipes cross and

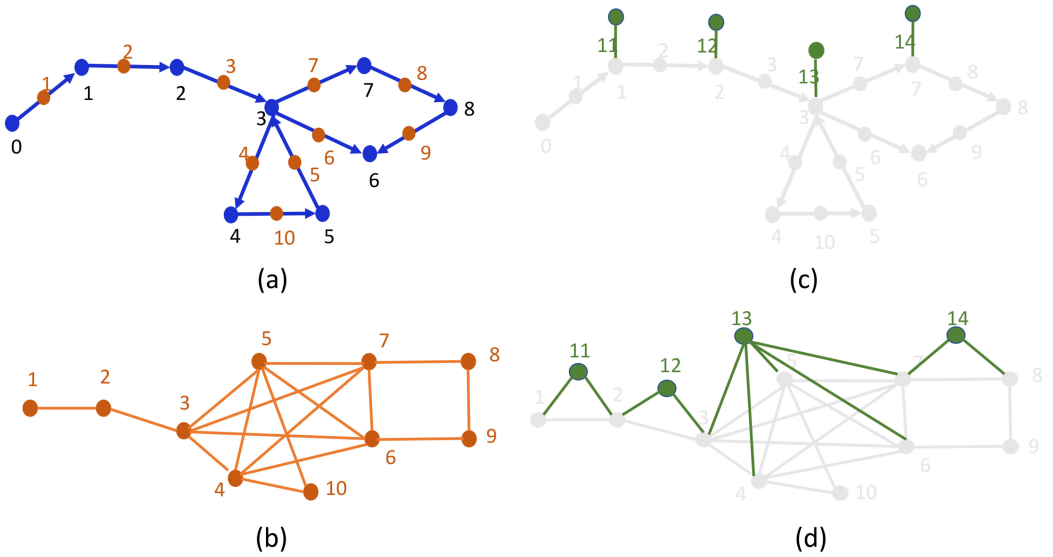


Fig. 3. Example of a line graph representation. In subfigure (a) we have the original directed graph, with $N = 9$ nodes and $N_e = 10$ links. The orange numbers associated to each link enumerates graph edges and they visually constitute the nodes for the line graph. In subfigure (b) we have the corresponding line graph made by N_e nodes. Each node correspond to a link of the original graph. In subfigure (c) we have the extension to the demand values. Indeed, some graph nodes of the original graph require water from the network. It is modeled by an outgoing virtual flow. The virtual flows in original graph generates virtual nodes in the line graph, that are represented in subfigure (d).

that can be usually equipped by sensors [27]. This graph carries information about the physical layout and functionality of the WDN. Indeed, the graph nodes correspond to junctions, essential components of the distribution system. The edges model the physical connections, typically pipes, allowing the water flow between different points in the network. The graph reflects the topological intricacies of the water distribution network, providing a mathematical representation that aids in understanding the network's architecture.

Together with the binary adjacency matrix $\mathbf{A}$ that structurally models the possible presence of a pipe between junctions of a WDN, we introduce a weighted version of the adjacency matrix $\mathbf{A}_{FLOW}$, where each element is scaled by the water flow f associated to that specific junction. Advanced techniques, such as **Signal on Graph (SoG)** analysis, can be employed within this framework to learn complex representations of the WDN. SoG methods have the capability to capture spatial and temporal dependencies, offering a more sophisticated understanding of the network's behavior and enhancing the accuracy of predictive models.

In Figure 3(a) we have an example of a graph with $N = 9$ nodes and $N_e = 10$ edges (both represented in blue) that will be used in the following subsection to present the approach.

3.2 Line Graph model Transformation and Node Ranking

In this subsection, we will present the following steps of the GraphSmart method presented in (Figure 2), that involves a graph transformation and the identification of the node ranking.

With the aim to identify an optimal edge monitoring strategy, we work on an alternative domain where the flow values are associated to a node rather than a link, and the theoretically grounded reconstruction techniques are available [28]. Specifically, we resort to the line graph $\mathcal{G}_L = (\mathcal{V}_L, \mathcal{E}_L)$

associated to the original network graph $\mathcal{G}$. Each node $v \in \mathcal{V}_L$ of the line graph $\mathcal{G}_L$ is associated to one and only one link $e \in \mathcal{E}$ in the original graph $\mathcal{G}$; besides, two nodes $v_1, v_2 \in \mathcal{V}_L$ are connected if and only if the corresponding links $e_1, e_2 \in \mathcal{E}$ of the original graph originate from a common vertex $v \in \mathcal{V}$ [29–31]. The proposed model is based on measurements on the links (pipes of the water network) that were available from the hydraulic model and simulator, specifically water flow measurements. In this context, the line graphs offer different theoretical advantages, such as the fact that the known and unknown data appear as signal at the nodes, and this can be exploited for flow estimation or measurements ranking. Fundamentally, the intrinsic physical constraints among the flows at the input/output of a physical junction are naturally and compactly written in terms of the line graph topology. We report a toy case example of how to derive a line graph in Figure 3(b). Here we represent line graph made by N_e nodes correspondent to the original graph in subfigure (a). It is possible to visually verify that each link in (a), that is enumerated with orange points, corresponds to a node (orange node) in (b).

After associating the original link and its flow value with a node in the line graph, we can apply GSP to the line graph and reconstruct the flows from a limited set of measurements. Indeed, in the original graph representation of a WDN, the information of interest corresponds to the hydraulic flow associated with each link (or pipe). However, to apply SoGs method at the node level, a line graph transformation of this original graph representation is needed. Specifically, this redefines the graph such that the nodes, instead of the links, become the primary carriers of the hydraulic flow information. This adjustment allows for the application of the SoG method directly on the nodes of the graph.

The adjacency matrix of the line graph $\mathbf{A}_L \in \mathbb{R}^{N_e \times N_e}$ depends on the adjacency matrix of the original graph $\mathbf{A}$:

$$\mathbf{A}_L = \mathbf{B}\mathbf{B}^T - 2\mathbf{I} \tag{1}$$

The incidence matrix $\mathbf{B}$ can be easily derived from the adjacency matrix since it represents just a different way to describe the interactions between graph nodes. Indeed, $\mathbf{B}$ contains a number of rows equal to the number of nodes of $\mathcal{G}$ and a number of columns equal to the number of existing edges N_e. It is a binary matrix, whose generic element b_{ij} is 1 if the node i is implied in the $j-th$ link, otherwise it is zero.

We now propose a node ranking strategy on the line graph, that corresponds to sorting the links (i.e., junctions) in the original graph. In detail, we rank the nodes of the line graph $\mathcal{G}_L$ according to their predictability from the neighbors. With this ranking, we can infer which flow values are straightforwardly predicted and which instead must be measured to reconstruct the flow with a targeted accuracy. The rationale behind the ranking is that central nodes can be predicted from their neighbors better than less interconnected ones.

Therefore, when only a reduced number of measurements can be collected, these should correspond to the least interconnected nodes of the line graph $\mathcal{G}_L$. Remarkably, these nodes in $\mathcal{G}_L$ correspond to less interconnected links of the original graph $\mathcal{G}$, such as links across different node clusters. This is coherent with recent research findings [32], indicating that such links significantly impact the network's algebraic properties. In line with this approach, we focus on node centrality [33] to capture its interconnections within the network.

We assess the centrality of the nodes $\mathcal{V}_L$ belonging to the line graph $\mathcal{G}_L$ using different centrality estimators [33], namely the "betweenness" $c_n^{(b)}$, the "closeness" $c_n^{(c)}$, and the "pagerank" $c_n^{(p)}$ centrality metrics.

The "*betweenness*" centrality metric $c_n^{(b)}$ of the n-th node $v_n \in \mathcal{V}_L$ depends on how many times this latter is included in the shortest paths $\mathcal{P}_{ij} = \{v_i, \ldots, v_j\}$, $i, j = 0, \ldots N-1$ between each and

every graph node pair $(v_i, v_j) \in \mathcal{V}_L$, and it is computed as follows:

$$c_n^{(b)} = \sum_{\forall (v_i, v_j) \in \mathcal{V}_L} \frac{|\{\mathcal{P}_{ij} \; s.t. \; v_n \in \mathcal{P}_{ij}\}|}{|\mathcal{P}_{ij}|}$$

where $|\mathcal{S}|$ denotes the cardinality of the set $\mathcal{S}$.

The "*closeness*" centrality metric $c_n^{(c)}$ of the n-th node $v_n \in \mathcal{V}_L$ depends on its distance from each and every other graph node $v_j \in \mathcal{V}_L, \; j \neq i$, and it is computed as follows:

$$c_n^{(c)} = \frac{1}{\sum\limits_{\forall (v_j) \in \mathcal{V}_L, j \neq i} d_{ij}}$$

where d_{ij} is the distance between nodes i and j.

The "*pagerank*" metric $c_n^{(p)}$ of the n-th node $v_n \in \mathcal{V}_L$ depends on how many times this latter is included in random walks. This algorithm determines the centrality based on how much each node appears within the shortest paths of the network.[1] $\mathcal{W}_{ij} = \{v_i, \ldots, v_j\}$, $i, j = 0, \ldots N-1$ between each and every graph node pair (i, j), $i, j = 0, \ldots N-1$

$$c_n^{(p)} = \sum_{\forall (v_i, v_j) \in \mathcal{V}_L} \frac{|\{\mathcal{W}_{ij} \; s.t. \; v_n \in \mathcal{W}_{ij}\}|}{|\mathcal{W}_{ij}|}$$

3.3 Green Reconstruction Algorithm

In this subsection, we introduce our proposed reconstruction algorithm, graphically represented as the last step of Figure 2. We recognize that within the system, there may be *unmeasured nodes*. These are nodes in the network that lack corresponding measurements or data points. This absence of data can occur for various reasons: either no sensor has been installed at these nodes, or the existing sensors are inactive, possibly due to being in sleep mode or having depleted batteries

According to the ranking proposed in the previous subsection phase, the measures at the most central nodes (highlighted by red circles) are not collected but reconstructed from adjacent, relevant nodes. This is illustrated in the right subfigure of Figure 2, where active sensors correspond to nodes 1,4,7,9, while unmeasured nodes are 2,3,5,6,8,10.

It means that we discard measurements at the most central nodes in $\mathcal{G}_L$, while keeping those at the least central ones, which are less predictable from their neighborhood. The centrality based ranking will identify the nodes that can be discarded without affecting the flow reconstruction accuracy.

In order to develop the reconstruction algorithm, we resort to the well-established WDS model adopted in the literature, e.g., in the Hardy-Cross method [34]. In our GraphSmart model, we assume that the water flow follows a distribution pattern where every junction adheres to the principle of continuity. The continuity equation dictates that the algebraic sum of flow rates in pipes converging at a node, along with any external flows, must be zero. These flows must meet the continuity requirement at every junction, i.e., the algebraic sum of the flow rates in the pipes connecting a junction, together with any demand flows, is zero. We have a condition on the water flow that means that the sum of inflows is equal to the outflows for every node in the graph if we assume the absence of leakage:

$$\mathbf{A}_{\mathbf{FLOW}} \cdot \underline{1} = \underline{0} \tag{2}$$

[1]The random walks are generated by selecting the successor of a node with probability 0.85 out of its neighbors and with probability 0.15 among all the other remaining network nodes.

From Equation (1) it is possible to rewrite this condition in terms of incidence matrix $\mathbf{B_{FLOW}}$ computed from $\mathbf{A_{FLOW}}$:

$$\mathbf{B_{FLOW}} \cdot \underline{1} = \underline{0} \tag{3}$$

A different perspective can be obtained by considering the incidence matrix of the physical graph $\mathbf{B}$, i.e., when the adjacency matrix $\mathbf{A}$ is the binary symmetric matrix that represents the presence of a pipe between two nodes. In this manner, the condition expressed in Equation (3) can be formulated as follows:

$$\mathbf{B} \cdot \underline{f} = \underline{0} \tag{4}$$

We assume to know only N_{e^K} values of $\underline{f}$, that represents the water flows, while the remaining N_{e^U} are the unmeasured nodes. For the sake of clarity, we identify two components $\underline{f}^K$ and $\underline{f}^U$ that correspond respectively to the known and unknown values of the signal $\underline{f}$.

$$[\mathbf{B}^U|\mathbf{B}^K]\begin{bmatrix}\underline{f}^U\\ \underline{f}^K\end{bmatrix} = \underline{0}$$

where $\mathbf{B}^U$ and $\mathbf{B}^K$ are the columns of the matrix $\mathbf{B}$ that are multiplied for $\underline{f}^U$ and $\underline{f}^U$, respectively. Let us remark that all the elements of $\mathbf{B}$ are known since they correspond to the geometry of the connections on the network. Following the development of the equations, it is possible to obtain a mathematical expression to compute $\underline{f}^U$, i.e., the unknown values of the SoG representing the water flow in the network:

$$\underline{f}^U = \mathbf{B}^{U+} \cdot \mathbf{B}^K \cdot \underline{f}^K \tag{5}$$

where $\cdot^+$ is the pseudo-inverse of the original matrix.

Extention to the case of water demands. The original WDS network model encompasses infrastructure elements and consumer endpoints. The associated line graph allows us to effectively estimate the flow within the network using a reduced number of sensors. Let us now delve into the representation of consumer endpoints. Each consumer endpoint involves an incoming flow, i.e., the water supplied by the network, and an outgoing flow, i.e., the water demanded by the consumer, that we refer to as demand value. The incoming water flow is associated to the physical connection of the endpoint with the infrastructure, and hence to a network edge. The outgoing flow is not directly tied to a physical connection. This notwithstanding, we can represent it by introducing in $\mathcal{G}$ an additional edge that connects the endpoint to a virtual boundary node, which does not correspond to a physical point of the network. This extended graph $\mathcal{G}_{\restriction \S \sqcup}$ contains N_e connections related to the physical pipes and N_e associated to demand values. With this extension, the adjacency matrix of the physical graph including demand values is $A_{ext} \in \mathbb{R}^{2N\times 2N}$:

$$A_{ext} = \left[\begin{array}{c|c} A & I \\ \hline 0 & 0 \end{array}\right] \tag{6}$$

where I is the identity matrix $I \in \mathbb{R}^{N\times N}$.

With this considerations, the adjacency matrix can be organized in blocks where we recognize the original adjacency matrix A, that models the physical structure of the network.Then we have another block $I \in \mathbb{R}^{N\times N}$ that reflects the possibility that each of the N nodes has a virtual outflow towards virtual boundary node. All the other elements of A_{ext} are zero since the virtual nodes cannot be connected in other ways.

With this position, the reconstruction algorithm applies regardless of whether there are outgoing flows at the consumer endpoints. Indeed, we seamlessly handle the water flows exiting the network at the consumer endpoints just like the flows within the network. These outgoing flows may be known (e.g., spilled out water from consumer measured from utility meter) or unknown

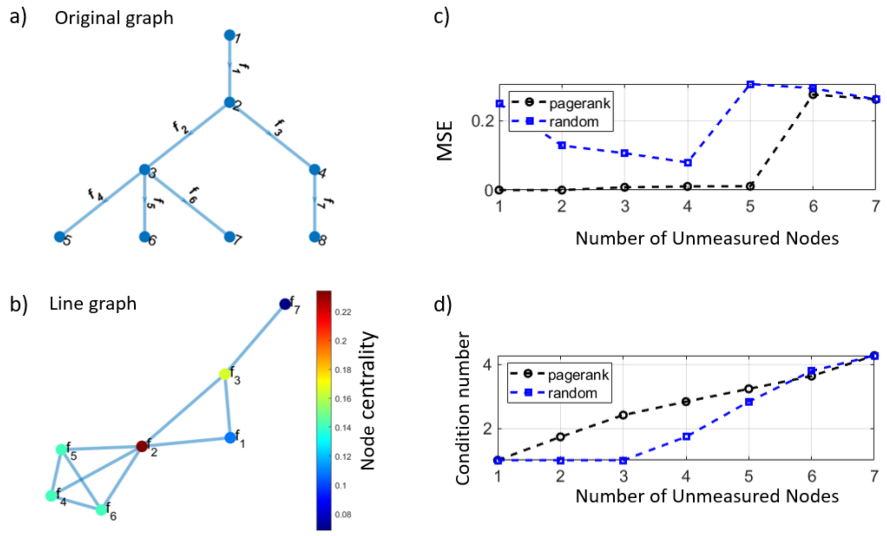


Fig. 4. Example use case representation. In (a) we have the original graph represented by graph links. In (b) we have the corresponding line graph. The color scale associated with the nodes of the line graph corresponds to their centrality, as measured by the PageRank algorithm. In (c) we report the MSE between ground truth and reconstructed flow values of the line graph as a function of the number of unmeasured sensors. In the black line, the unmeasured nodes are ranked according to their descending centrality, while in the blue line they are randomly sorted. In (d) we have the condition number for inversion, which measures the quality of the inversion in Equation (7).

(e.g., presence of leakage in some point of the WDS infrastructure), depending on the specific observed WDS network. We graphically show the presence and the effect of the addition of virtual nodes in Figure 3(c). We have the green links in subfigure (c) representing the endpoints flow, and the associated line graph Figure 3(d), where the corresponding nodes are also highlighted in green in the transformed line graph.

4 GraphSmart Algorithm Evaluation

In this section, we focus on realistic WDN networks and we study the accuracy of the reconstruction, quantified by the **Mean Square Error (MSE)** between the ground truth $\underline{f}_j, j = 0, \dots N-1$ and the reconstructed flow $\hat{\underline{f}}_j, j = 0, \dots N-1$, computed as: $\sum_{j=0}^{N-1}(\hat{\underline{f}}_j - \underline{f}_j)^2$. We relate the accuracy to the number of available sensors, and we show that the approach leads to energy efficient monitoring system. We present the performance of the proposed method in two different cases.

In order to better understand the key ideas and the algorithm steps, the first case shows results on an example network with a limited number of links, with and without demand values. Then, the second case presents the results of the proposed approach in a complex WDN scenario, extracted from a real infrastructure and obtained by the EPANET tool, which is a software able to generate WDN with real physical constraints associated with the water flow.

4.1 Results on Example Use Case

In this subsection, we provide results from a case study with the objective of understanding the functionality of the proposed approach. This analysis numerically illustrates the different stages of the proposed algorithm within a controlled scenario, as depicted in Figure 4. Specifically, we consider the graph of a WDN with $N = 8$ nodes $1, 2..N$ and $N_e = 7$ edges $f_1, f_2, ..., f_{N_e}$ depicted in

Figure 4(a). Then, we derive the associated line graph as in Equation (1) that is made by N_e nodes, where each node corresponds to a link in the original graph, represented in Figure 4(b). The node centrality of the line graph is calculated using the pagerank algorithm as outlined in Section 3. We represent the node centrality by the color of the nodes in Figure 4(b), with color scale levels indicated in the figure legend. As illustrated in the figure, the node with the highest centrality value is situated at the center of the graph ($N_e = 2$), which corresponds to node $N = 2$ in the original graph.

According to the proposed approach, this node is a strong candidate for removal as it represents a location where the flow can be more readily reconstructed. Our key hypothesis is that the first nodes, i.e., the most central nodes, are less important in the reconstruction algorithm.

In order to test this idea, we start to eliminate the values associated with the first ranked node and we apply the reconstruction algorithm. After the reconstruction algorithm, we compute the MSE between the reconstructed and ground truth node signal value. Let us remark that this operation corresponds in the original graph, to the reconstruction of the unmeasured water flow values through the others.

We replicate these steps after the elimination of other node values according to their centrality and we obtain the results in the black line of Figure 4(c). Results show that we can switch-off the five most central nodes and perfectly reconstruct their values, that means that only the two less central nodes, i.e., f_7 and f_1 (depicted in blue according to the color scale), are needed to completely recover the node signals.

To assess whether the ranking has an impact on the algorithm's ability to reconstruct the network, we replicate the same procedure but with a random ranking of the nodes (blue line in the figure). In this scenario, the performance deteriorates because there is a higher probability of removing nodes with lower centrality first. This indicates that the order of node removal, based on their centrality, plays a significant role in the successful reconstruction of the network flow.

Finally, in Figure 4(d) we report the condition number for inversion C defined as $C(\mathbf{B}^U) = ||\mathbf{B}^U|| \cdot ||\mathbf{B}^{-1U}||$. This metric is frequently used in literature to gauge the reliability of algorithms, particularly those involving matrix inversions [35]. A small condition number for inversion implies that the problem is well-conditioned, meaning the solution is stable and reliable. Conversely, a large condition number signifies an ill-conditioned problem. In such a case, minor changes in the input can lead to significant changes in the output, making the solution less reliable and more sensitive to input variations.

In our applications, a large C means that the quality of the inversion of the matrix $\mathbf{B}^U$ is poor. The analysis of the condition number shows that when all the node measurements, corresponding to the water flows, are available, C is low, meaning that the inversion of $\mathbf{B}^U$ is well performed. As we eliminate node values and reconstruct them through the others, the condition number increases and the problem becomes ill-conditioned making impossible the correct reconstruction of the node values.

To assess the case where physical demands are present in the network, we now perform the analyses in a modified version of the previous graph, in which we have additional edges representing demand values, i.e., the possibility that nodes of the network have additional outflows. This is represented in Figure 5(a). According to the proposed model, anticipated in Section 3, we represent the water demand as an additional outgoing flow from each node of the network that demands water to the network. The virtual edges representing demand values are in green (f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}) in Figure 5(a) and they connect physical nodes, i.e., 1, 2, 3, 4, 5, 6, 7, 8, with virtual additional nodes, i.e., 9, 10, 11, 12, 13, 14.

From the original graph with those extensions, we derive the line graph in Figure 5(b) according to Equation (1) and we compute the node centrality with pagerank algorithm. The node centrality

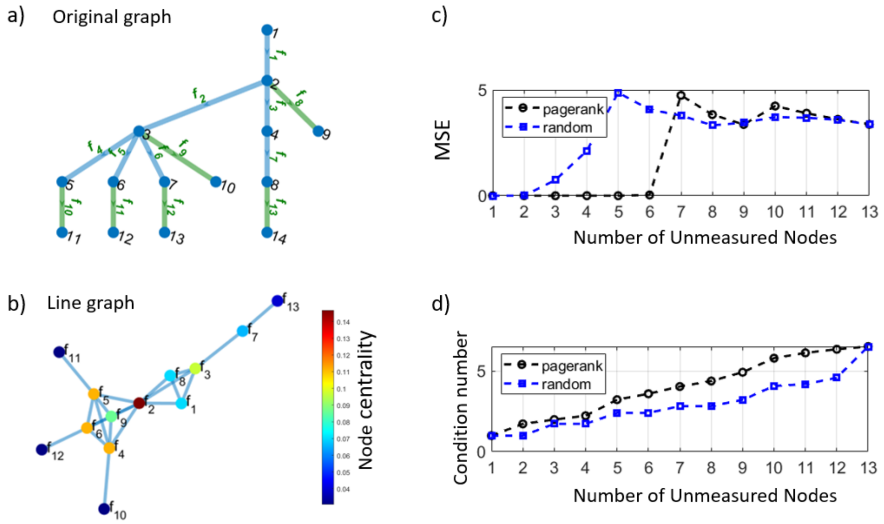


Fig. 5. Example use case representation in the presence of demand values. In (a) we have the original graph represented by graph links coloured in blue, and demand values, represented by additional nodes and links colored in green. In (b) we have the associated line graph. The color associated to the nodes of the line graph corresponds to their centrality, as measured by the pagerank algorithm. In (c) we report the MSE between ground truth and reconstructed node values of the line graph as function of the number of unmeasured nodes. In the black line, the unmeasured nodes are ranked according to their descending centrality, while in the blue line, they are randomly sorted. In (d) we have the condition number for inversion, which measures the quality of the inversion in Equation (7).

is represented as the color of the nodes. Following the same approach explained for the previous analysis, we derive the MSE between the reconstructed and the ground truth node values as a function of the number of unmeasured nodes, reported in Figure 5(c). Our results show that the reconstruction error if we properly rank the nodes of the line graph remains low (i.e., almost zero) for a larger number of unmeasured nodes as compared to the absence of ranking. Specifically, we can eliminate 6 over 13 node values and correctly recover the signal.

Let us remark the insertion of demand values has caused a decrease in the percentage of eliminable node values. Indeed in the absence of demand values, we can switch-off a percentage of 71% of total nodes while in the presence of demand values, this percentage falls to 46%. This point is in line with the model since we do not change the topology of the network but we just have additional virtual nodes that relate to the original nodes of the graph and, consequently, the number of umeasuring nodes that we can allow for a specific topology does not significantly change.

4.2 Results on Realistic WDN

To test GraphSmart in a realistic infrastructure, we utilized a pre-defined network model extracted from a real infrastructure and available in the Open Water Analytics community public repository [36]. We consider EPANET and the **Water Network Tool for Resilience (WNTR)** [37] to generate our desired hydraulic data. WNTR is a Python package designed to simulate real-time WDN. WNTR interfaces EPANET, which is an open source software to model hydraulic and quality dynamics of a WDN [38]. EPANET takes into account the topological structure of the pipeline system along with a set of initial conditions (e.g., pipe diameter) and rules of how the system is operated so that it can compute flows and pressures throughout the network for a specific period of time. The simulator can add leaks to the network using a leak model [39].

Table 4. Attributes of the Pipes

Element	Description	Unit
hour	A timestamp representing the time-interval we are currently watching in the simulation	H:M:S
linkID	Unique ID of a link inside the network	Not applicable
start_node	The source node of the link	Not applicable
end_node	The target node of the link	Not applicable
flowrate	The flow rate of the water inside the pipe at the current timestamp	GPM: gal/min
velocity	The velocity of the water inside the pipe at the current timestamp	ft/s

The flow direction within these pipes is determined by the difference in hydraulic head (the internal energy per weight of water or pump effect), with water flowing from the end with higher hydraulic head to the end with lower head.

Table 5. Attributes of the Nodes

Element	Description	Unit
hour	A timestamp representing the time-interval we are currently watching in the simulation	H:M:S
nodeID	Unique ID of a node inside the network	Not applicable
demand	Rate of water withdrawal from the network. A negative value is used to indicate an external source of flow into the junction	GPM: gal/min
head	Hydraulic head (i.e., elevation + pressure head) of water in the node of the WDN	ft
pressure	Measured pressure in the node of the WDN	psi
x_pos,y_pos	Coordinates of the node	meters
node_type	A string which tells the type of the node (i.e., “Junction”, “Reservoir”, “Tank”)	Not applicable

Each node can be configured with a specific base demand pattern which represents the water request of the user during the whole simulation changing at a step size of an hour.

The tools analyze the geometric structure of a pipeline system, taking into account various initial conditions such as pipe roughness and diameter, as well as operational rules. By doing so, it is able to calculate flows, pressures, and water quality parameters (such as disinfection concentrations and water age) throughout the network for a specific time period.

By using these tools we generate a suitable dataset to apply the proposed method. To facilitate data analysis, we have chosen to export physical WDN values evolving during the time into two **“Comma-Separated Values" (CSV)** files, the first one which is related to the WDN junctions (nodes) and includes the physical objects that constitute the distribution system as well as its running parameters, and the second one which is related to the WDN pipes (links). Table 4 and Table 5 contain the comprehensive set of features and fields associated with the links and nodes, respectively, as reflected in the provided dataset that is publicly available on our GitHub page [10].

Specifically, our focus was on the networks illustrated in Figure 6. This network comprises a total of $n = 83$ junctions and $m = 1$ reservoir. Each node within the network can be customized with a specific *base demand* pattern, representing the water requirements of the users throughout the simulation. The demand pattern changes at an hourly interval and is evenly distributed within a given range. In this study, we examined two scenarios: the first scenario excluded demand configuration for the nodes, allowing water flow throughout the network while preventing any water spillage. The second scenario involved configuring the nodes to have a demand evenly distributed within a specified range.

The figure employs two color scales to represent the demand values of the nodes (indicated by the color of the circle markers) and the flow within the pipes (indicated by the color of the connection segments between nodes). Both were measured in GPM: gal/min. The reservoir, identified by the label “7384," is positioned on the left side of the network, with the predominant flow direction being from left to right. As we move away from the reservoir, both the total demand

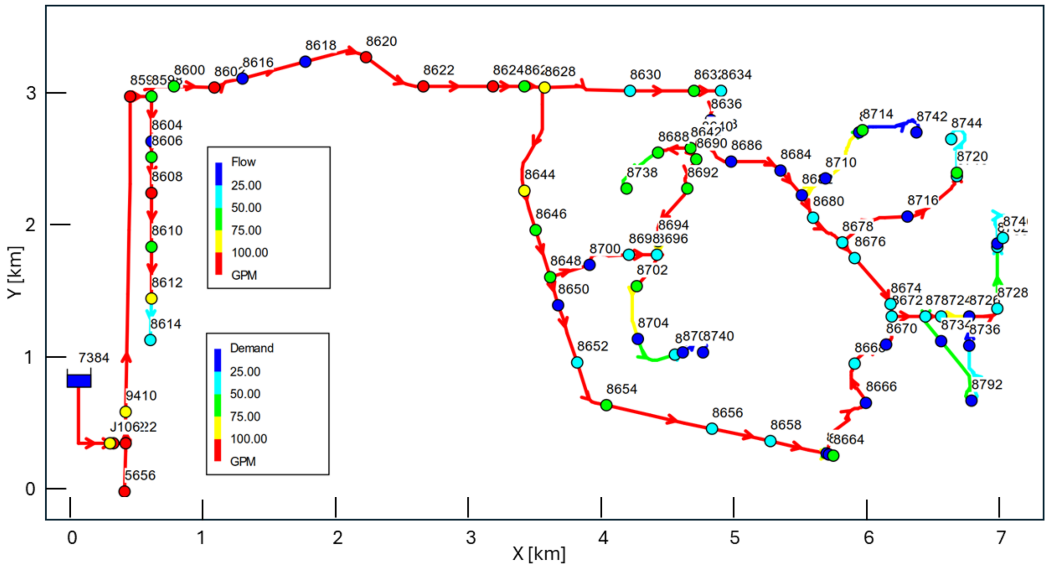


Fig. 6. Selected realistic WDN branch with 83 junctions and one reservoir. Junctions demands and pipes flows measurements are represented in the figure by color depicted in the two legends scale.

value and the pressure at the junctions decrease due to the demand from the junctions on the left side. The links correspond to pipes and they are represented with an arrow in order to show the direction of the water flow.

We conducted a simulation encompassing a one-month running experiment. To ensure the repricability of our experiments, the dataset of this experiment can be found in the publicly available repository provided in [10].

We now apply the proposed method to this realistic network. We firstly derive the binary adjacency matrix whose generic element $\mathbf{A}_{ij}$ is equal to 1 if there is a link connecting the node i to the node j. Then we derive the incidence matrix $\mathbf{B}$ and the adjacency matrix of the associated line graph $\mathbf{A}_L$ as in Equation (1) and rank the nodes in this transformed space according to their centrality, measured by the pagerank algorithm.

In Figure 7(a) we have the line graph associated to the realistic network, where the color of each node corresponds to their centrality. It is interesting to focus on the least central nodes, represented in blue, which correspond to physically isolated nodes and are the most important nodes for reconstruction.

In Figure 7(b) we report the MSE between reconstructed and ground truth node values as function of the number of unmeasured nodes in two cases, i.e. the ranking with pagerank centrality and random sorting. Our findings show that by properly choosing the correct sorting for eliminating the node values it is possible to have until 73 over 85 unmeasured nodes. In other words, it means that we can place only 12 sensors on this network made by 85 nodes and we can completely reconstruct the network. We remark that if we randomly choose the sequence of unmeasured nodes, the MSE severally increases when we eliminate only 37 sensor values.

We now consider the presence of demand values, that means that we add the possibility that nodes demand water from the network. We compute the adjacency matrix in its extended expression $\mathbf{A}_{ext}$ in Equation 6. In this case, the total number of links is equal to 168, and consequently, the line graph in the presence of demand values has 168 nodes, shown in Figure 8.

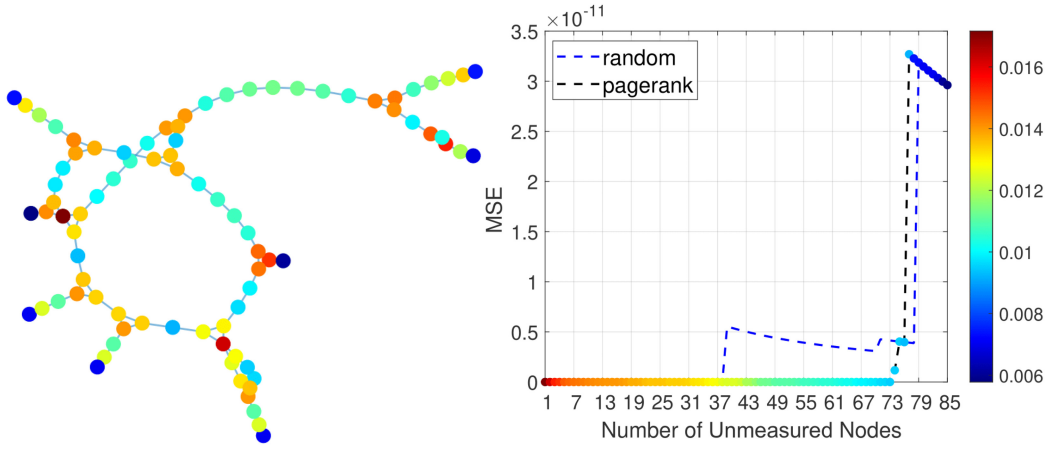


Fig. 7. Results of GraphSmart in the EPANET realistic network. In subfigure (a) we have the line graph where the color of each node corresponds to its centrality, measured by pagerank algorithm. In subfigure (b) we report the MSE between reconstructed and ground truth node values as a function of the number of unmeasured nodes in two cases, i.e., the ranking with pagerank centrality (in black line with nodes highlighted according to their centrality) and random sorting (in blue).

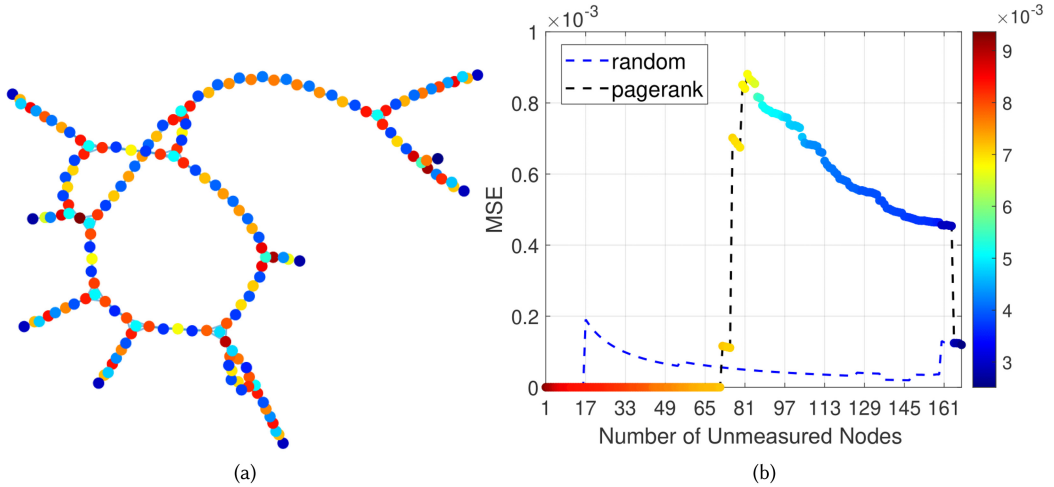


Fig. 8. Results of GraphSmart to the EPANET realistic network in presence of demand values. In subfigure (a) we have the line graph where the color of each node corresponds to its centrality, measured by pagerank algorithm. In subfigure (b) we report the MSE between reconstructed and ground truth node values as function of the number of unmeasured nodes in two cases, i.e., the ranking with pagerank centrality (in black line with nodes highlighted according to their centrality) and random sorting (in blue).

We rank the nodes of the line graph according to their centrality and we represent it as color of the nodes in Figure 8(a). We replicate the MSE analysis in this scenario and we prove that our ranking enables us to have 71 unmeasured nodes over the 168 proposed locations, instead of 17 by randomly eliminating node values.

A few remarks on the presented results are in order. Firstly, adding virtual links (and nodes) to model water demand values does not change the graph topology and for this reason, the number

of nodes that are unmeasured, given the topology of the network, does not significantly change in the presence of demands. Secondly, the MSE is a step function, meaning that when the configuration of available node values enables the algorithm to reconstruct the water flow, the MSE is close to zero. Then, as soon as a critical node for the reconstruction is removed the reconstruction algorithm breaks and the MSE is on the same order of magnitude of the water flow. This consideration motivates the oscillations of the MSE around the maximum value since the reconstruction algorithm completely fails.

4.3 GraphSmart in the Presence of Perturbations

The results in the previous sections relate to the underlying assumption of deterministic measurements acquired by IoT devices. Here, we consider an extension to the case of noisy measurements with the aim to investigate the robustness of the approach in a more realistic scenario.

To this aim, we consider a perturbed scenario where measurements are affected by an additive Gaussian noise at a different **Signal-to-Noise Ratio (SNR).** In this case, the measured flows $\hat{\underline{f}}$ differ from $\underline{f}$ for the normally distributed estimation error: $\hat{\underline{f}} = \underline{f} + \underline{w}$, with $\underline{w}$ being Gaussian vector at a defined SNR.

In this context, the reconstruction formulation can be rewritten as follows:

$$\hat{\underline{f}}^U = \mathbf{B}^{U+} \cdot \mathbf{B}^K \cdot \hat{\underline{f}}^K \tag{7}$$

We now consider the WDN in Figure 6 with additive Gaussian noise on water flows at different SNR values: SNR = 30db, 15db, 10db, to then apply the GraphSmart method. In Figure 9 we report the reconstruction error as function of the number of unmeasured nodes N_{e^U}.

We firstly prove the robustness of our algorithm by demonstrating that additive noise until 30 dB does not change the curve with respect to that obtained in absence of noise, in Figure 7(c). This high level of noise tolerance indicates that the algorithm is able to guarantee high accuracy in noisy environments. We now consider the cases of SNR = 10, 15 db: we remark that after a first MSE increase caused by the presence of high noise, it remains stable until it is not possible to correctly reconstruct the water flows. Also in the noisy scenario, MSE exhibits a step-wise pattern, indicating that the reconstruction error remains relatively stable by increasing the number of unmeasured nodes, until a threshold is reached. Then, the reconstruction algorithm fails, and the error increases to levels similar to those of the flow values.

These characteristics are crucial for real-world applications, especially in environments commonly affected by noise or similar impairments, as they ensure the algorithm can be actually applied.

5 IoT Monitoring Scenario for GraphSmart

Our approach is based on an IoT system of water sensors that are placed at the pipe of the WDNs to collect measurements of the flow in the network.

In terms of the cost of LoRaWAN meter devices, there is a significant variation of prices ranging from 50 to several hundreds of dollars. However, in our study, we are specifically referring to devices that are installed in the backbone pipelines of WDNs, where also the installation and maintenance processes could be complex.

Considering the growing emphasis on energy efficiency, the graph network can be harnessed to suitable design of sensor placement and data transmission, thereby minimizing energy consumption. The GraphSmart approach preserves the accuracy of network reconstruction while lowering energy consumption. With the aim of achieving this objective, the process involves the identification of critical nodes that offer essential information for flow reconstruction, while simultaneously

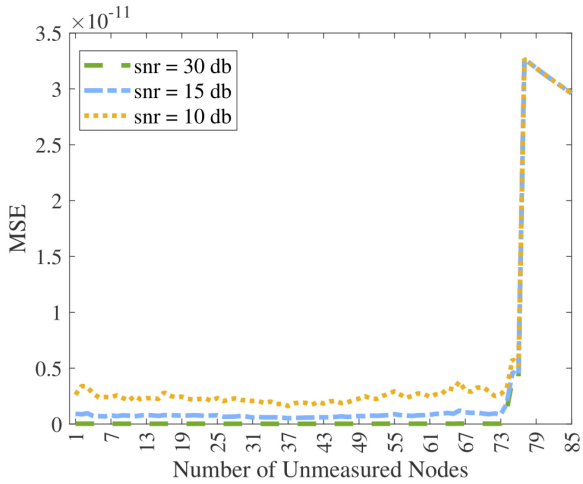


Fig. 9. MSE varying the number of unmeasured nodes (N_eU) for different signal-to-noise-ratio (SNR) values.

minimizing energy consumption by sensors. To evaluate the GraphSmart approach in a LoRaWAN network communication scenario, we thoroughly examine its energy-related characteristics.

From the power consumption aspect, an ideal IoT chip needs to have very low power consumption during operation, ultra-low deep sleep current, and long-distance communication capability. The advent of LoRaWAN technology solves the problem of transmission distance, network signal and power consumption, and is the most cost-effective, which is exactly what we are looking forward to in the field of water meter reading applications [7]. Especially, in this work we focus on a flow meters deployment.

In the subsequent sections, we begin by presenting the definition of an energy model for the LoRaWAN flow meter device. Subsequently, we introduce the chosen simulation framework for conducting our performance evaluation. Building upon this selected simulation framework, we establish a precise scenario that faithfully aligns with the WDN deployment discussed in the preceding section, as well as the defined energy model.

LoRaWAN utilizes LoRa modulation, a Chirp Spread Spectrum modulation to facilitate long-distance, low data rate communication within the sub-1GHz ISM bands. Communication between sensors and GWs is spread out on different frequency channels and data rates. LoRaWAN uses up to six different programmable **Spreading Factors (SFs)**: 7, 8, 9, 10, 11, 12. Furthermore, the adopted bandwidth can be configured: 125 kHz, 250 kHz and 500 kHz (typically 125 kHz for the 868 ISM band). LoRa devices use a higher SF when the signal is weak or there is strong interference in the used channel. For instance, if a device is located at a considerable distance from a GW, the signal becomes weaker and thus requires a higher SF. Utilizing a higher SF results in a prolonged symbol duration, and consequently, a longer ToA. This has a notable impact on power consumption. Indeed, higher device energy consumption is associated with a higher SF, while lower consumption is observed with a lower SF. This difference is mainly due to the transmission duration of each device, which becomes more substantial when the SF is higher. The choice of the SF represents a trade-off between communication range and packet duration or power consumption. The task of selecting the optimal SF value lies with the network server, facilitated by the **Adaptive Data Rate (ADR)** module. In this study, we focus on identifying the selection strategy for sensor meters within the already deployed WDS network, which is characterized by the presence of only two GWs. These gateways have been strategically placed to provide optimal coverage for all the existing devices.

Table 6. LoRaWAN Scenario Simulation Parameters for the Realistic WDN Considered in this Work

Parameter	Value
Carrier Frequency (MHz)	863.0
Bandwidth (kHz)	125
Code Rate	4/5
Message size [byte]	20
Message Period	1 packet every 300 seconds
Number of gateway	2
Number of nodes	85
TXPower	14 dBm
Path loss values	$\eta = 2.9$, $\sigma^2 = 0$, $\overline{L_{pl}}(40m) = -66\ dB$

Computer modeling and simulation are valuable methods for exploring system performance and evaluating strategies without costly implementation. Simulation becomes particularly valuable in IoT scenarios characterized by numerous nodes and large geographical areas. In the context of LoRaWAN networks, simulation enables the design and evaluation of LoRa-based applications prior to actual deployment. While various LoRaWAN simulation tools exist in the literature, a study referenced in [40] identifies the NS-3 network simulator as the most suitable option.

NS-3 not only supports LoRaWAN technology but also offers extensive libraries [41]. NS-3 is an open-source discrete-event network simulator written in C++ and Python. The NS-3 simulator supports a wide variety of protocols such as Wi-Fi, LTE, IEEE 802.15.4, SigFox, LoRaWAN, and further networks.

The integrated LoRaWAN module in NS-3 is designed to meet the requirements of Class A devices. Class A devices are known for being the most power-efficient among the three available classes in LoRaWAN (Class A, Class B, and Class C).

LoRaWAN NS-3 module considers N LoRaWAN sensors placed in a 2-dimensional space around M gateways. In the simulator, the **Receiver Signal Strength Indicator (RSSI)** value, associated with the sensor, depends on the distance between the device and the gateway according to the $L_{pl}(d)$ path loss model:

$$L_{pl}(d) = \overline{L_{pl}}(d_0) + 10\eta \log\left(\frac{d}{d_0}\right) + \chi_\sigma \quad [dB] \tag{8}$$

where $\overline{L_{pl}}(d_0)$ is the mean path loss at the reference distance d_0, η is the path loss exponent and $\chi_\sigma \sim \mathcal{N}(0, \sigma^2)$ is the normal distribution with zero mean and σ^2 variance to account for shadowing.

Ultimately, we activate the channel capture effect to create a more realistic scenario. LoRa [42], like all orthogonal modulations when different SFs are used, is highly resistant to Gaussian noise as well as self-interference caused by colliding transmissions at the same SF. Indeed, in the event of collisions among two or more transmitters, an even small difference of power, is sufficient to correctly demodulate the strongest colliding signal [43]. This phenomenon, known as "channel capture," has a profound impact on the scalability of LoRa technology, as deploying multiple GWs can substantially increase the capture probability, thereby enhancing the overall network capacity.

This assumption is reasonable when the cell works in stable conditions, and collisions involving multiple overlapping frames are rare or have a dominant contribution in the interfering power as introduced in [44], [45].

Unless otherwise specified, Table 6 shows the scenario parameters used in our simulations. To support the selection of a suitable path loss model, which includes a representative value for the

Table 7. Current Consumption of the Devices in Different Operating Modes

Device	Transmit	Receive	Sensing	Sleep
FiPy	170 mA	120 mA	110 mA	50 μA
TTGO	111 mA	61 mA	51 mA	30 μA
STM32WL55JC	28 mA	11 mA	1.4 mA	1.5 μA

propagation coefficient, we considered the study in [46]. Based on their findings, the value of η equal to 2.9 was chosen, as it is the most representative value among those listed in the study cited above.

LoRaWAN encompasses three categories of end devices, each with distinct characteristics. Class A is specifically designed for metering applications and consumes minimal energy and it is considered for this study. Thus, we do not consider **downlink (DL)** transmissions, which involve messages sent from gateways to end devices. As a result, sensor devices do not receive downlink data. We believe this limitation is relatively minor, given that the majority of the traffic in the selected scenario will be in the **uplink (UL)** direction.

In our research, we run experiments within a simulation tool to assess performances across different selected rankings. By profiling real device energy consumption, we accurately depict the energy usage of individual device components. We focus on a LoRaWAN Class A network configuration, characterized by end devices always initiating transmissions in a completely asynchronous manner. One of the key system parameters in our scenario is the reporting periodicity denoted as T, where each end device is allocated a unique initial reporting delay, following which it generates a new packet every T seconds.

To maximize both the battery life of sensors and overall network capacity, LoRaWAN can manage the data rate and SF output for each sensor individually by means of an ADR scheme [47]. This mechanism determines the transmission parameters (SF and transmit power) of the device based on the estimation of the link budget in the uplink and the threshold RSSI for decoding the packet correctly at the current data rate. When the data rate is not achievable the ADR reduces it (by increasing the SF) to provide connectivity to the device. Finally, the network will be optimized to use the fastest data rate possible for each sensor.

In the applied scenario, at the onset of the simulation, each sensor receives an SF assignment according to the following procedure. Initially, we calculate the power level received by each gateway from the sensor. Subsequently, we select the gateway that records the highest received power and determine the SF based on that power level. In this assignment, we ensure that the sensor is assigned the lowest SF that surpasses the sensitivity of the gateway. We refer to this procedure as ADR in the next part of our work. The goal is to optimize communication between the sensor and the gateway while maintaining efficient data transmission.

For what concerns the energy model, a LoRaWAN library for Class A in NS-3 has been previously developed by [48], [49], and [50]. In this research, we perform an extensive study of LoRaWAN sensors in terms of energy consumption. We conducted field experiments to evaluate the current consumption in the four operating modes (sensing, transmitting, receiving, and sleeping). In particular, we estimated the power consumption of three devices, which include the SX1276 LoRa transceiver. We used a Tektronix MSO 2024B oscilloscope with TCP0020 current probe to measure the current absorbed by the sensors, and we set supply voltage at 3.3 V, devices transmission power was set at 25 mW (14 dBm).

The average current consumption of the three considered sensors in all the four mentioned states is outlined in Table 7. In particular, the table shows that in the three active states (transmit,

receive, sensing), the power consumption of the STM32WL55JC device is lower than the FiPy and TTGO devices. The reason for this difference is that the first two are systems specifically designed for prototyping. As such, they include additional components that enhance consumption. After evaluating the energy consumption characteristics of the devices, the decision was made to concentrate the analysis exclusively on the STM32WL55JC device [51]. Indeed, this ultra-low power device is highly suitable for large-scale green IoT deployment [52]. Thus, we use the results to adapt the available energy consumption module in NS-3 to enable the evaluation of LoRaWAN networks in terms of energy efficiency.

6 GraphSmart Integrated Performance Evaluation

In this section, we present the advantage results of the GraphSmart approach, where we include the energy aspect, on the synthetic WDN presented in Section 4.2. To this goal, we leveraged the NS-3 simulator module and conducted a series of assessments to gauge the performance metrics of the IoT network. These tests encompassed evaluations of network performance, GWs coverage and energy consumption.

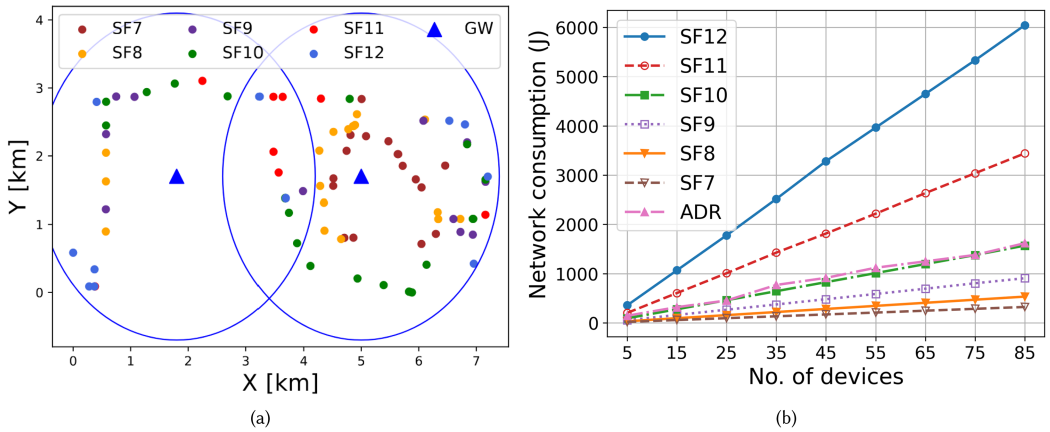
6.1 WDN LoRaWAN Monitoring Scenario

The primary objective of the initial simulation campaign was to assess the impact of the selected SF on energy consumption. Energy consumption was determined by monitoring the battery level of the devices, initially set at 10,000 joules. Each device transmitted data at an average interval of 5 minutes and the battery level of each device was observed over a 24-hour experimental period. The exact interval time was uniformly distributed within the range of 4 to 6 minutes, with the value extracted at the end of each transmission. Finally, we calculated the total energy expended across the entire LoRaWAN network by summing up the energy consumption of the network devices.

Our focus in this study revolves around monitoring the flow rate within each pipe. To achieve this, we position a sensor at the end of every link. As a result, the number of devices exceeds the number of junctions, since multiple pipes may terminate at the same junction. In such cases, multiple devices are positioned at the same location to monitor the flow at the end of multiple pipes. In the end, we placed 85 monitoring LoRaWAN devices.

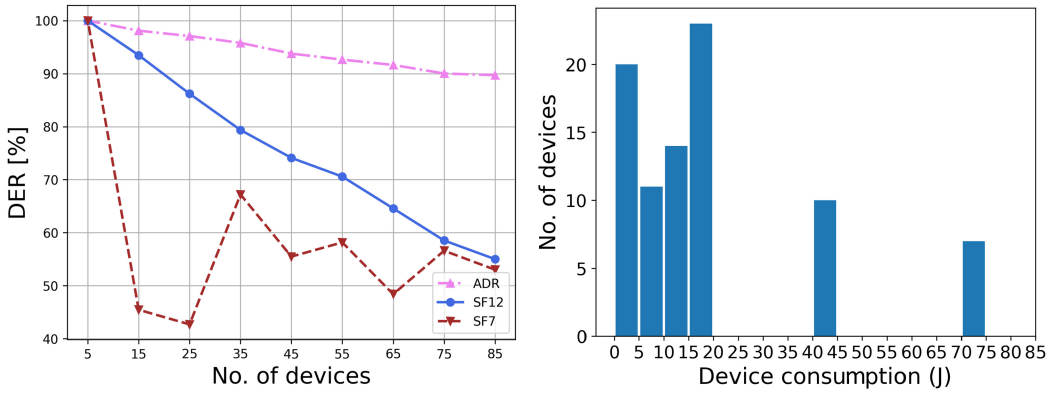
Lastly, based on the chosen network topology, we determine the optimal placement of GWs to efficiently collect the monitoring data. The considered GWs positions are shown in Figure 10(a) (triangular markers are the GWs) together with the coverage area when SF=12 (maximum coverage area) is configured (represented by big blue circles). According to the coverage area, each GW is able to receive only a subset of the total devices measurements. Figure 10(a) also shows the location of the nodes and the SF values assigned by the ADR. For instance, the devices represented by the black dots had an SF of 7 and were placed near the GWs, while it assigns higher SF values to devices farther from the GWs, such as the devices represented by the yellow dots, which had an SF of 12. In the figure, the legend exactly depicts the color assigned to each SF. Furthermore, Figure 10(b) further analyzes the network's energy consumption as a function of the number of inactive or undeployed devices. This analysis considers scenarios where Adaptive Data Rate (ADR) is either enabled or disabled. When ADR is disabled, all devices are configured to use the same spreading factor (SF). Multiple simulations were conducted with devices using a fixed SF ranging from 7 to 12 and with ADR enabled. The objective was to evaluate the impact of the selected SF on the energy efficiency of the entire network. In order to conduct the experiment, we randomly removed devices following their initial order, with a regular interval of five steps.

Higher network energy consumption was observed with $SF12$ configuration for the devices, while lower consumption was observed with $SF7$; this is due to the transmission duration of each



(a) (b)

Fig. 10. Devices deployment and SF values assigned by ADR in the proposed realistic scenario (a). Total energy consumption in joule of the network as a function of the number devices present (b).



(a) (b)

Fig. 11. DER as a function of the number of EDs in three representative scenarios, enabled ADR, all devices configured with SF=12 and SF=7 (a). Frequency distributions of devices numbers based on their consumption in the presence of ADR (b).

device, predominant when the SF is higher. The line depicted by triangular markers corresponds to the scenario where ADR is enabled. In this case, the SF configuration of the devices is set as shown in Figure 10(a). It should be noted that this particular scenario does not represent the optimal energy efficiency. However, it is important to observe that low consumption with SF set to 7 (optimal energy efficiency) does not necessarily imply optimal network efficiency. Indeed, it is also crucial to evaluate the packet delivery ratio.

To this end, we extract the probability of correctly receiving a packet, which is a typical metric considered for characterizing LoRaWAN systems (often called **Data Extraction Rate - DER**) of the main three configured scenarios. Specifically, Figure 11(a) illustrates the trend of DER as a function of the number of devices in the network in three different scenarios: in the presence of active ADR (up triangular markers), when all devices are set to SF7 (down triangular markers), and SF12 (circular marker). Based on these configurations, it can be observed that the DER parameter for SF7 and SF12 drops below 60% when the two gateways serve more than 75 devices. Both scenarios

negatively impact the performance of the devices, albeit for different reasons. In the case of SF=7 configuration, a significant number of devices are left out of coverage, resulting in a lower number of packets being received from gateways. On the other hand, when SF=12 is configured, the longer duration of the packets leads to multiple collisions, thereby reducing the probability of successful reception.

Conversely, with ADR active, the DER remains above or equal to 90%, allowing for a maximum number of serviceable devices up to 85. By comparing Figure 10(b) and Figure 11(a), results indicate that the ADR increases data delivery success within the network while maintaining an acceptable energy consumption, thereby increasing network energy efficiency.

To better visualize the consumption of the network devices, Figure 11(b) shows the frequency distribution of network devices according to their 24-hour energy consumption in the presence of ADR. Specifically, it is evident that the majority of devices within the studied network exhibit a daily energy consumption of less than 20 joules, with these devices predominantly utilizing lower SF values. Conversely, a subset of devices, approximately 18 out of the total 85, exhibit higher daily energy consumption, attributed to their configuration with SF values of 11 and 12. We note that in LoRaWAN, the duration of a packet doubles as the SF value increases by one unit. This exponential increase in duration is the reason behind the corresponding exponential increase in power consumption.

The introduction of ADR once again demonstrated notable improvements in energy efficiency. For instance, if all devices had been consistently configured with an SF of 12, the entire set of 85 devices would have incurred an approximate daily energy consumption of 71 joules per device, corresponding to 6,035 joules when considering the entire network. As a result, it can be concluded that the adoption of ADR represents the optimal trade-off between energy consumption and network throughput in this context. Nevertheless, energy efficiency can be further enhanced through the GraphSmart approach, as discussed in the following subsection.

6.2 GraphSmart Energy Saving

This subsection presents a comprehensive analysis of the four ranking strategies presented in Section 3.3. This analysis focuses on evaluating the performance of the GraphSmart strategies in terms of both flow reconstruction MSE and energy efficiency. The goal is to identify the most effective optimization strategy for selecting an optimal sensor deployment in the WDN scenario.

We consider the MSE as a measure of reconstruction performance. We have demonstrated in Section 4.2 that it performs as a step function, whose values rapidly change in a critical condition when a necessary node value is unmeasured. To stress this point, we report in Figure 12 the sensor deployment configuration at the critical step. It means that the presented figure is the configuration just before the algorithm breaks. We have the line graph in subfigure (a) and the original graph in subfigure (b). In blue we have nodes corresponding to active sensors while in red we have unmeasured values to be reconstructed. It is interesting to remark that the following removed link, that breaks the algorithm, is the 8644 – 8646. We remark that this condition reflects the optimal sensor deployment suggested by the algorithm because it corresponds to the minimum set of active sensors that maintains the reconstruction accuracy and minimizes the energy consumption.

We compare the results of four ranking strategies and their impact on the two key metrics: MSE and energy consumption. By evaluating the performance of each strategy, we aim to provide insights into the most suitable approach for achieving an optimal deployment. To facilitate the analysis, we present a figure that visually depicts the results of the MSE and energy consumption for each ranking strategy.

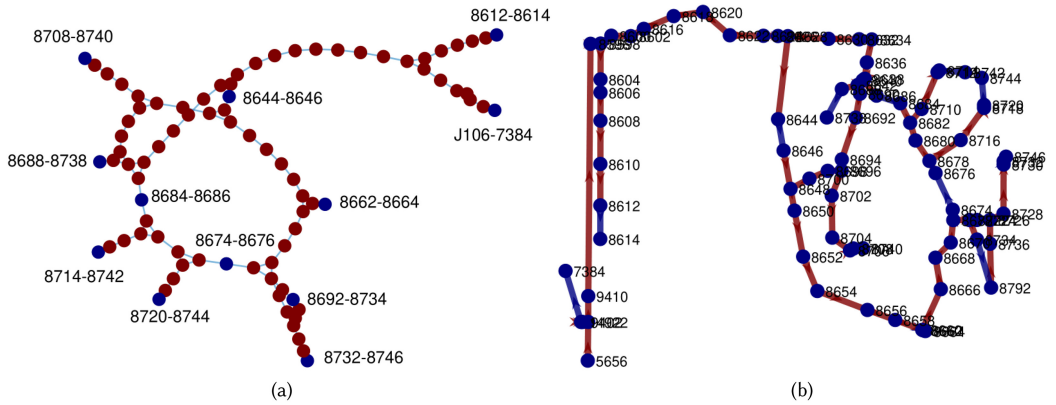


Fig. 12. The sensor deployment configuration at the critical step. In subfigure (a) we have the line graph associated to the EPANET realistic network. In red we have unmeasuring nodes (i.e., inactive or sensors that can be not deployed at all) while in blue we have nodes corresponding to active sensors. Links needed to reconstruct the signal are identified through the start and end node. In subfigure (b) we report with the same colour code the configuration in the original graph and we observe the link position of the essential nodes (represented in blue) for the reconstruction algorithm.

Figure 13 illustrates a comparison between the MSE of flow reconstruction and energy consumption for the four rankings, *pagerank*, *closeness*, *betweeness* and random. Two subplots sharing the same x-axis are represented in the figure. The first subplot shows the MSE error of the network flow reconstruction versus the network energy consumption; the MSE is expressed as a logarithmic scale and the error is reduced when the energy consumption increases. The network energy consumption information is reported in the sharing axis.

The second subplot provides the number of unmeasuring devices as function of the energy consumption. We remark, once again, that those sensors can be deployed but inactive or not deployed at all. Intuitively, the energy consumption is high when few sensors are removed, while it decreases when the number of unmeasured nodes increases. Both the subplots show four lines, one for each ranking modality applied to remove sensors from the network.

With the aim to better understand the energy-reconstruction performance of GraphSmart we can discuss Figure 13 in its entirety. Starting from the first subplot, we can fix a tolerated value for the MSE and we can point to the correspondent logarithmic value (as shown by the arrows) in the top figure. Then, we can derive the network energy consumption associated with the fixed error value for the specific ranking procedure that we are considering. Finally, we can read the number of sensors that can be turned off on the y-axis of the bottom figure in correspondence of the derived network consumption.

Our findings show that pagerank is the method that produces the best-sorted devices list. Indeed, after fixing the value of the tolerated error, the pagerank sorting algorithm enables us to turn off the largest number of devices. This indicates that using pagerank as an optimization strategy for choosing the optimal sensors deployment in the WDN scenario yields the best results in terms of both MSE and energy consumption. In particular, Table 8 highlights that the pagerank algorithm stands out as the best in terms of energy savings. This ranking is capable of reconstructing the water network flow with only 20 installed devices and remarkable energy savings of 73.01%.

Taken together, we demonstrate that GraphSmart is an effective approach for optimizing the deployment of resources in a WDN, and it is able to balance flow reconstruction accuracy and energy efficiency.

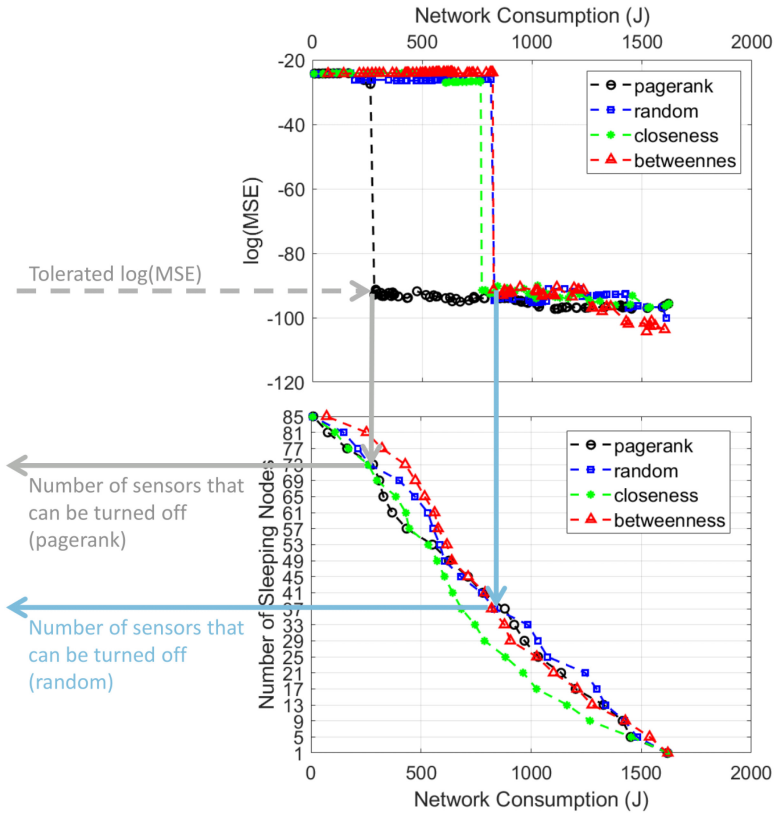


Fig. 13. Integrated results of efficient and accurate GraphSmart. The top subfigure shows the MSE in logarithmic scale vs. the energy consumption of the network. The bottom subfigure presents the number of sensors that can be removed as function of the network energy consumption. The two figures share the same x-axis and they can be read together. We can select the tolerated MSE (represented with the dot line) and derive the network consumption associated to the centrality selected algorithm. Following the arrows on the left we obtain the number of sensors that can be removed to obtain the desired reconstruction error. In grey, we have the pagerank and in blue, the random cases.

7 On the Scalability and Computational Complexity of GraphSmart Reconstruction Algorithm

7.1 Scalability

In order to test the scalability of the proposed GraphSmart method, we test it on a large WDN, made by $N = 815$ nodes and $N_e = 1125$ pipes, that is represented in Figure 14(a). We replicate all the steps of the algorithm by firstly deriving the line graph in Figure 14(b). Then, we compute the MSE between reconstructed values and ground truth in order to quantify the ability of the approach to correctly monitor the water distribution network in this large scenario. Our findings show that the MSE increases as the number of unmeasured nodes increases in a similar way to Figure 7(c). Indeed, we report a step-wise behavior of the MSE according to which it is possible to derive the desired number of nodes that can be inactive while keeping the needed reconstruction accuracy. For the sake of comparison, we report the MSE results obtained for a random selection of nodes with the blue line of Figure 7(c). From the observation of the random node strategy for the reconstruction, we conclude that the optimal ranking of nodes strongly impact the network

Table 8. Ranking Comparison with Associated Energy Saving Percentages

Ranking	Installed sensor devices	Energy saving (%)
Random	48	48.81
Betweenness	48	49.35
Closeness	48	58.11
Pagerank	20	73.01

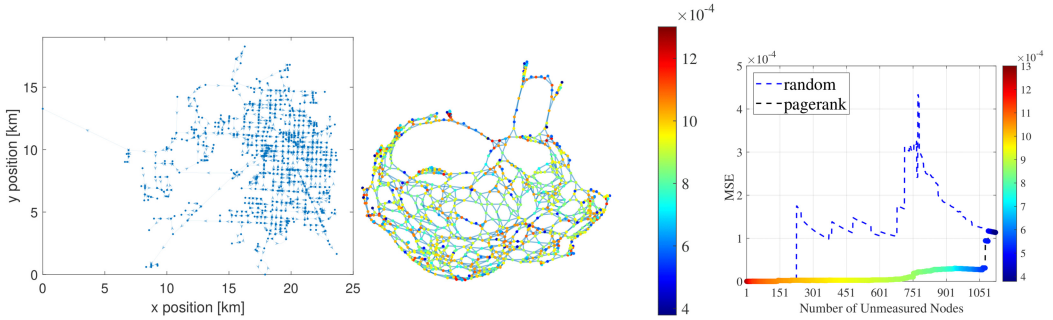


Fig. 14. Results of GraphSmart to the realistic network with $N = 815$ and $N_e = 1125$. In subfigure (a) we have the original graph with the scale of the deployment in km. In subfigure (b) we have the line graph where the color of each node corresponds to its centrality, measured by pagerank algorithm. In subfigure (c) we report the MSE between reconstructed and ground truth node values as function of the number of unmeasured nodes in two cases, i.e., the ranking with pagerank centrality (in black line with nodes highlighted according to their centrality) and random sorting (in blue).

monitoring and the high dimensionality of the WDN requires a careful node selection method, which is well achieved with pagerank centrality ranking.

7.2 Computational Complexity

The computational complexity of the GraphSmart reconstruction algorithm, that is written in the expression: $\underline{f}^U = \mathbf{B}^{U+} \cdot \mathbf{B}^K \cdot \underline{f}^K$, depends on:

- the pseudo-inverse computation of the matrix $\mathbf{B}^U$, that is $o(N^2 \cdot N_{e^u})$, assuming that N$\leq N_{e^u}$, otherwise it is $o(N \cdot N_{e^u}^2)$;
- the matrix multiplication between $\mathbf{B}^{U+}$ and $\mathbf{B}^K$ is $o(N_{e^u} \cdot N \cdot N_{e^k})$;
- the vector-matrix multiplications, that is $o(N_{e^u} \cdot N_{e^k})$.

The dominant term of the computation complexity of the GraphSmart algorithm is the pseudoinverse of $\mathbf{B}^{U+}$. In order to investigate the behavior of this term, we represent it in the plot behind that corresponds to the new plot of Figure 15

This figure shows the behavior of the computational complexity of the dominant term (the pseudo-inverse) for a WDN consisting of $N = 815$ nodes and $N_e = 1125$ links. The entire figure is divided into two regions by a vertical dashed line at $N_{e^u} = N = 815$. The area to the left of this line is shaded in orange, indicating the complexity region where the number of unmeasured nodes (N_{e^u}) of the line graph $\mathcal{G}_L$ (N_{e^u}) is less than the number of nodes N, while the area to the right is shaded in light green, representing the complexity region where the number of unmeasured nodes is $N_{e^u} > N$. This figure represents the mathematical shape of the computational cost of

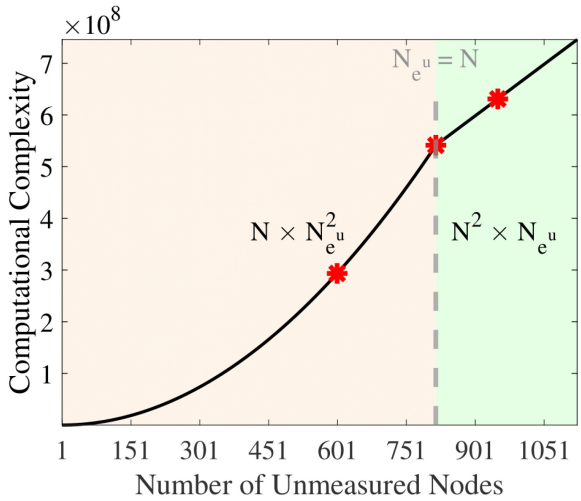


Fig. 15. Computational complexity varying the number of unmeasured nodes.

Table 9. Computational Time for Three Conditions of $N_{e^u} < N$, $N_{e^u} = N$ and $N_{e^u} > N$, that Respectively Correspond to $N_{e^u} = 600$, $N_{e^u} = 815$, and $N_{e^u} = 950$, that are the Points Associated to the Red Asterisks in Figure 15

WDN	$N_{e^u} < N$	$N_{e^u} = N$	$N_{e^u} > N$
$N = 815$, $N_e = 1125$	0.22 s	0.54 s	0.58 s

The computational times are obtained with Intel(R) Core(TM) i5-7200U CPU @ 2.50 GHz, 2712 Mhz, 2 cores, 4 logical processors.

the dominant term of GraphSmart but we also compute the computational time for the points represented with red asterisks in the plot and we report them in the following table.

Table 9 reports the computational time for $N_{e^u} = 600$, $N_{e^u} = 815$, and $N_{e^u} = 950$, that respectively corresponds to $N_{e^u} < N$, $N_{e^u} = N$, and $N_{e^u} > N$.

Analyzing the results in Table 9, it is possible to conclude that even in the presence of a network made by more than a thousand of flow values to be reconstructed, computational times are below or equal to 500ms (computational times are obtained with Intel(R) Core(TM) i5-7200U CPU @ 2.50 GHz, 2712 Mhz, 2 cores, 4 logical processors). This is compatible with reconstruction operations that can be implemented by water operators to accurately monitor the WDN. Let us remark that it is always possible to cluster the network in order to analyze in parallel the sub-networks with the aim to further improve the computational time.

8 Conclusions and Future Work

The main purpose of this work is to introduce a novel method to optimize the monitoring of Water Distribution Systems (WDSs) by an effective IoT sensor placement. WDSs consist of physical pipes enabling the water flow and IoT sensors measuring features such as flow rates, pressure levels, and water quality parameters. The hydraulic connectivity between nodes and pipes can be modeled as a graph by possibly adding other physical properties, such as the resistance or the diameter of pipes. In this scenario, sensor measurements acquired by an IoT network can be modeled as signals defined in a networked domain.

In this paper, we analyze the line graph associated with the physical graph and we introduce a graph signal processing-based approach to reconstruct the flow using a reduced number of

sensors. Specifically, we took into consideration several node centrality measures for comparison's sake and we demonstrate the pagerank centrality computed in the line graph is able to optimally identify which sensors are not needed and can be switched off. In comparison with the alternative metrics, pagerank enables the accurate reconstruction of the water flow with the minimum set of IoT measures. The proposed approach, named GraphSmart, is tested with numerical simulations performed in a realistic LoRaWAN WDS scenario. We demonstrate that the water flow is accurately reconstructed by strategic placement of a reduced number of sensors, leading to significant energy savings, i.e., around 73%. The MSE shape also constitutes a way to practically determine the number of sensors that have to be necessarily activated or deployed in the network: if only a subset of the nodes are selected for flow reconstruction, starting from the least central ones and in order of increasing centrality, the GraphSmart MSE intrinsically exhibits a step-like behavior, so that for a given number of measurements the MSE drastically diminishes; for a larger number of measurements, the MSE still diminishes, with a much slower slope. This behavior, observed on networks of different sizes, and on different noise conditions, is the reason for identifying an MSE-based threshold. Specifically, the threshold can be selected as the number of nodes for which a slope discontinuity is observed.

Our study faces real-world challenges in the sensor deployment. Indeed, the LoRaWAN deployment scenario influences factors such as battery life or application scale. With the aim to manage those critical aspects, we use the Adaptive Data Rate (ADR) modules and, in future work, strategic gateway placements will be investigated. We remark that we tested our algorithm on several WDNs and GraphSmart reconstruction algorithm always finds a reduced subset of required sensors. This enhances network capacity by reducing measurement points, which corresponds to a decreased network traffic and limited maintenance needs. Our monitoring strategy is resilient to measurement losses since it ensures robustness in reconstructing unmeasured points.

The GraphSmart method has been applied to a realistic network, assuming that the IoT measurements are repeated at a pace related to the WDS stationarity interval. In future work, we can further reduce the number of IoT sensors by designing joint subsampling of the measurements both in the time and space domains. This can be accomplished by a multilayer vertex-time representation of the WDS network, where it is possible to jointly select samples in the vertex and time domain. This would extend the analysis on GraphSmart sensor displacement to identify the time resolution of IoT devices readings. Future work will address the analysis of a multilayer network representing the physical flow constraints along time, in order to identify the location and timing of IoT readings and improve the WDS flow reconstruction accuracy. From the perspective of the wireless network, our plan is to incorporate the optimization of GW deployment into the approach presented in this work.

Taken together, the GraphSmart method offers versatile and insightful mapping of water distribution networks. By encompassing topology, hydraulic characteristics, and advanced optimization techniques, our proposed approach contributes to a holistic understanding of the network's complexities, supporting endeavors to enhance efficiency, accuracy, and energy conservation within water distribution systems. In conclusion, GraphSmart represents a potentially useful tool for water operators that currently face problems related to water scarcity.

References

[1] David Seckler, Randolph Barker, and Upali Amarasinghe. 1999. Water scarcity in the twenty-first century. *International Journal of Water Resources Development* 15, 1-2 (1999), 29–42.

[2] Junguo Liu, Hong Yang, Simon N. Gosling, Matti Kummu, Martina Flörke, Stephan Pfister, Naota Hanasaki, Yoshihide Wada, Xinxin Zhang, Chunmiao Zheng, Joseph Alcamo, and Taikan Oki. 2017. Water scarcity assessments in the past, present, and future. *Earth's Future* 5, 6 (2017), 545–559.

[3] I. UNEP. 2016. Options for decoupling economic growth from water use and water pollution. *Report of the International Resource Panel Working Group on Sustainable Water Management*.
[4] Directorate-General for Environment European Commission. 2019. Report from the Commission to the European Parliament and the Council on the implementation of the Water Framework Directive (2000/60/EC) and the Floods Directive (2007/60/EC). (2019).
[5] Zheng Liu and Yehuda Kleiner. 2013. State of the art review of inspection technologies for condition assessment of water pipes. *Measurement* 46, 1 (2013), 1–15.
[6] Antonino Pagano, Domenico Garlisi, Fabrizio Giuliano, Tiziana Cattai, and Francesca Cuomo Sapienza. 2024. SWIFEED: Smart water IoT framework for evaluation of energy and data in massive scenarios. In *2024 IFIP Networking Conference (IFIP Networking)*. IEEE, 583–585.
[7] Mohammed Rezwanul Islam, Sami Azam, Bharanidharan Shanmugam, and Deepika Mathur. 2022. A review on current technologies and future direction of water leakage detection in water distribution network. *IEEE Access* 10 (2022), 107177–107201. DOI: http://dx.doi.org/10.1109/ACCESS.2022.3212769
[8] LoRa Alliance Technical Committee. 2017. LoRaWAN 1.1 Specification. (October 2017). https://lora-alliance.org/resource-hub/lorawantm-specification-v11 Accessed: 2020-04-16.
[9] SigFox. https://build.sigfox.com/sigfox-device-radio-specifications
[10] GraphSmart applicated dataset. ([n. d.]). https://github.com/WITS-Restart/WDN-Dataset Accessed: 2023-10-28.
[11] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M. F. Moura, and Pierre Vandergheynst. 2018. Graph signal processing: Overview, challenges, and applications. *Proc. IEEE* 106, 5 (2018), 808–828.
[12] Georgios B. Giannakis, Yanning Shen, and Georgios Vasileios Karanikolas. 2018. Topology identification and learning over graphs: Accounting for nonlinearities and dynamics. *Proc. IEEE* 106, 5 (2018), 787–807.
[13] Nitin J. Williams, Ian Daly, and Slawomir J. Nasuto. 2018. Markov model-based method to analyse time-varying networks in EEG task-related data. *Frontiers in Computational Neuroscience* 12 (2018), 76.
[14] G. V. Karanikolas and Georgios B. Giannakis. 2017. Identifying directional connections in brain networks via multikernel Granger models. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'17)*. IEEE, 6304–6308.
[15] Weiyu Huang, Thomas A. W. Bolton, John D. Medaglia, Danielle S. Bassett, Alejandro Ribeiro, and Dimitri Van De Ville. 2018. A graph signal processing perspective on functional brain imaging. *Proc. IEEE* 106, 5 (2018), 868–885.
[16] Alexandre Ciancio, Sundeep Pattem, Antonio Ortega, and Bhaskar Krishnamachari. 2006. Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*. 309–316.
[17] Armando Di Nardo, Michele Di Natale, Giovanni F. Santonastaso, Velitchko G. Tzatchkov, and Victor H. Alcocer-Yamanaka. 2014. Water network sectorization based on graph theory and energy performance indices. *Journal of Water Resources Planning and Management* 140, 5 (2014), 620–629.
[18] S. Mohan Kumar, Shankar Narasimhan, and S. Murty Bhallamudi. 2008. State estimation in water distribution networks using graph-theoretic reduction strategy. *Journal of Water Resources Planning and Management* 134, 5 (2008), 395–403.
[19] Xiao Zhou, Shuming Liu, Weirong Xu, Kunlun Xin, Yipeng Wu, and Fanlin Meng. 2022. Bridging hydraulics and graph signal processing: A new perspective to estimate water distribution network pressures. *Water Research* 217 (2022), 118416.
[20] Armando Di Nardo, Carlo Giudicianni, Roberto Greco, Manuel Herrera, and Giovanni F. Santonastaso. 2018. Applications of graph spectral techniques to water distribution network management. *Water* 10, 1 (2018), 45.
[21] Velitchko G. Tzatchkov, Victor H. Alcocer-Yamanaka, and Víctor Bourguett Ortíz. 2008. Graph theory based algorithms for water distribution network sectorization projects. In *Water Distribution Systems Analysis Symposium 2006*. 1–15.
[22] Manuel Herrera, Edo Abraham, and Ivan Stoianov. 2016. A graph-theoretic framework for assessing the resilience of sectorised water distribution networks. *Water Resources Management* 30 (2016), 1685–1699.
[23] Daniel Bezerra, Rui Souza, Gustavo Meirelles, and Bruno Brentan. 2022. Leak detection in water distribution networks based on graph signal processing of pressure data. In *Int. Jt. Conf. Water Distrib. Syst. Anal*, Vol. 86. 14073.
[24] Aravind Rajeswaran, Sridharakumar Narasimhan, and Shankar Narasimhan. 2018. A graph partitioning algorithm for leak detection in water distribution networks. *Computers & Chemical Engineering* 108 (2018), 11–23.
[25] Jiheon Kang, Youn-Jong Park, Jaeho Lee, Soo-Hyun Wang, and Doo-Seop Eom. 2017. Novel leakage detection by ensemble CNN-SVM and graph-based localization in water distribution systems. *IEEE Transactions on Industrial Electronics* 65, 5 (2017), 4279–4289.
[26] Hafiz Husnain Raza Sherazi, Luigi Alfredo Grieco, Muhammad Ali Imran, and Gennaro Boggia. 2020. Energy-efficient LoRaWAN for industry 4.0 applications. *IEEE Transactions on Industrial Informatics* 17, 2 (2020), 891–902.
[27] Jonathan W. Berry, Lisa Fleischer, William E. Hart, Cynthia A. Phillips, and Jean-Paul Watson. 2005. Sensor placement in municipal water networks. *Journal of Water Resources Planning and Management* 131, 3 (2005), 237–243.

[28] Aamir Anis, Aly El Gamal, A. Salman Avestimehr, and Antonio Ortega. 2018. A sampling theory perspective of graph-based semi-supervised learning. *IEEE Transactions on Information Theory* 65, 4 (2018), 2322–2342.
[29] Lowell W. Beineke. 1970. Characterizations of derived graphs. *Journal of Combinatorial Theory* 9, 2 (1970), 129–135.
[30] Dragoš Cvetkovic, Peter Rowlinson, and Slobodan Simic. 2004. *Spectral Generalizations of Line Graphs: On Graphs with Least Eigenvalue-2*. Vol. 314. Cambridge University Press.
[31] Tim S. Evans and Renaud Lambiotte. 2009. Line graphs, link partitions, and overlapping communities. *Physical Review E* 80, 1 (2009), 016105.
[32] Elena Ceci and Sergio Barbarossa. 2020. Graph signal processing in the presence of topology uncertainties. *IEEE Transactions on Signal Processing* 68 (2020), 1558–1573.
[33] Francisco Aparecido Rodrigues. 2019. Network centrality: An introduction. *A Mathematical Modeling Approach from Nonlinear Dynamics to Complex Systems* (2019), 177–196.
[34] Hardy Cross. 1936. *Analysis of Flow in Networks of Conduits or Conductors*. Technical Report. University of Illinois at Urbana Champaign, College of Engineering. *Engineering Experiment Station*.
[35] Nicholas J. Higham. 2002. *Accuracy and Stability of Numerical Algorithms*. SIAM.
[36] OpenWaterAnalytics. ([n. d.]). https://raw.githubusercontent.com/OpenWaterAnalytics/epanet-example-networks/master/epanet-tests/large/NW_Model.inp Accessed: 2023-08-28.
[37] Katherine A. Klise, Michael Bynum, Dylan Moriarty, and Regan Murray. 2017. A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study. *Environmental Modelling & Software* 95 (2017), 420–431. DOI: http://dx.doi.org/10.1016/j.envsoft.2017.06.022
[38] Jun Liu and Yu Guoping. 2013. Iterative methodology of pressure-dependent demand based on EPANET for pressure-deficient water distribution analysis. *Journal of Water Resources Planning and Management* 139 (01 2013), 34–44. DOI: http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000227
[39] Daniel A. Crowl and Joseph F. Louvar. 2001. *Chemical Process Safety: Fundamentals with Applications*. Pearson Education.
[40] Mukarram A. M. Almuhaya, Waheb A. Jabbar, Noorazliza Sulaiman, and Suliman Abdulmalek. 2022. A survey on LoRaWAN technology: Recent trends, opportunities, simulation tools and future directions. *Electronics* 11, 1 (2022). DOI: http://dx.doi.org/10.3390/electronics11010164
[41] Davide Magrin, Stefano Avallone, Sumit Roy, and Michele Zorzi. 2021. Validation of the Ns-3 802.11ax OFDMA implementation. In *Proceedings of the 2021 Workshop on Ns-3 (WNS3 '21)*. Association for Computing Machinery, New York, NY, USA, 1–8. DOI: http://dx.doi.org/10.1145/3460797.3460798
[42] Semtech. 2015. LoRa. (2015). http://www.semtech.com/ Accessed: 2020-07-20.
[43] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello. 2018. Impact of LoRa imperfect orthogonality: Analysis of link-level performance. *IEEE Communications Letters* 22, 4 (April 2018), 796–799. DOI: http://dx.doi.org/10.1109/LCOMM.2018.2797057
[44] Domenico Garlisi, Ilenia Tinnirello, Giuseppe Bianchi, and Francesca Cuomo. 2020. Capture aware sequential waterfilling for LoRaWAN adaptive data rate. *IEEE Transactions on Wireless Communications* 20, 3 (March 2020), 2019–2033. DOI: http://dx.doi.org/10.1109/TWC.2020.3038638
[45] Domenico Garlisi, Stefano Mangione, Fabrizio Giuliano, Daniele Croce, Giovanni Garbo, and Ilenia Tinnirello. 2021. Interference cancellation for LoRa gateways and impact on network capacity. *IEEE Access* 9 (2021), 128133–128146. DOI: http://dx.doi.org/10.1109/ACCESS.2021.3108021
[46] Domenico Garlisi, Antonino Pagano, Fabrizio Giuliano, Daniele Croce, and Ilenia Tinnirello. 2023. A coexistence study of low-power wide-area networks based on LoRaWAN and Sigfox. In *2023 IEEE Wireless Communications and Networking Conference (WCNC'23)*. IEEE, 1–7.
[47] Dae-Young Kim, Seokhoon Kim, Houcine Hassan, and Jong Hyuk Park. 2017. Adaptive data rate control in low power wide area networks for long range IoT services. *Journal of Computational Science* 22 (2017), 171 – 178. DOI: http://dx.doi.org/10.1016/j.jocs.2017.04.014
[48] Floris Van den Abeele, Jetmir Haxhibeqiri, Ingrid Moerman, and Jeroen Hoebeke. 2017. Scalability analysis of large-scale LoRaWAN networks in ns-3. *IEEE Internet of Things Journal* 4, 6 (2017), 2186–2198. DOI: http://dx.doi.org/10.1109/JIOT.2017.2768498
[49] Brecht Reynders, Qing Wang, and Sofie Pollin. 2018. A LoRaWAN module for ns-3: Implementation and evaluation. In *Proceedings of the 10th Workshop on ns-3 - WNS3 '18*. ACM Press, Surathkal, India, 61–68. DOI: http://dx.doi.org/10.1145/3199902.3199913
[50] Davide Magrin, Marco Centenaro, and Lorenzo Vangelista. 2017. Performance evaluation of LoRa networks in a smart city scenario. In *2017 IEEE International Conference on Communications (ICC'17)*. IEEE, Paris, France, 1–7. DOI: http://dx.doi.org/10.1109/ICC.2017.7996384

[51] STMicroelectronics. Ultra-low Power Multi-modulation Wireless STM32WL5x Microcontrollers. Accessed: 17/05/2023. https://www.st.com/en/microcontrollers-microprocessors/stm32wl5x.html ([n. d.]).
[52] Roberto La Rosa, Lokman Boulebnane, Daniele Croce, Patrizia Livreri, and Ilenia Tinnirello. 2023. An energy-autonomous and maintenance-free wireless sensor platform with LoRa connectivity. In *2023 12th International Conference on Renewable Energy Research and Applications (ICRERA'23)*. IEEE, 461–464.

Received 10 November 2023; revised 13 June 2024; accepted 30 August 2024