

A distributed analysis of vibration signals for leakage detection in Water Distribution Networks

Gabriele Restuccia^{1,2,a}, Ilenia Tinnirello^{1,2,a}, Fulvio Lo Valvo^{1,a},
Giacomo Baiamonte^{1,a}, Domenico Garlisi^{1,2,a*}, and Costantino Giaconia^{1,2,a}

¹University of Palermo, Italy –

²CNIT, Parma, Italy

^a{name}.{surname}@unipa.it

Keywords: Smart Water Distribution Network (SWDN), LoRaWAN, Vibrations Sensor

Abstract. It is well known that Water Distribution Networks (WDNs) are very inefficient and, in Italy, 40% of water is lost during distribution. In this paper, we present a solution for detecting leakages in WDNs, based on three main components: i) an innovative sensing element to be deployed at the sensor nodes, which analyses vibrations in the acoustic range for classifying external noise sources, induced by water leakages, by means of suitable machine learning techniques; ii) an Internet of Things (IoT) system of sensors, deployed at the junctions of the WDNs, for comparing the measurements collected at different critical points of the network; iii) a machine learning algorithm for processing the data. After the definition of the WDN structure, we introduce some numerical simulation tools suitable for studying our system and modeling the proposed sensing solution. Given the geometry, physical properties (pipe lengths, diameters, roughness, reservoir shapes and levels, pump and valve characteristic curves) and nodal demands, the simulation tool is able to compute leakages in pipes or nodes over time. In parallel, we simulate our IoT system coupled to the WDN, by logging partial information about the WDN status, which corresponds to the demand readings at the edge nodes or at some junction nodes, together with the (optional) measurements of the deployed sensing elements. On the basis of this data, we analyze the possibility of identifying the leakages in the network, even without knowing the exact or complete topology of the WDN. Our solution exploits different machine learning techniques devised to indirectly retrieve topological information, by correlating the balance of the flows as the water demand varies over time.

Introduction

A Water Supply System (WSS) is the infrastructure that connects water suppliers and customers to provide water to households and businesses. A conventional WSS comprises a water source, a water treatment, a pumping station, a Water Distribution Network (WDN), and finally, the end users. Over the past 100 years, the global water consumption has increased sixfold and this trend has continued at a rate of about 1% per year due to population growth and to the economic development [1]. According to the current literature, the world may face a global water deficit of 40% by 2030. At the same time, massive amounts of water are wasted because of leakages in (aging) water distribution infrastructures. According to the more recent statistics of the European associations of water services (EurEau), the mean value for water wasted is about 26% in Europe [2], but such a value can be even higher than 50% in some networks. In this context, it is of paramount interest to exploit ICT solutions for improving the monitoring and management of WSSs. Indeed, most water utilities address the leakage control in a passive way; leaks are repaired only when they are visible. A standard way to facilitate the leakage control is to partition the network into District Metered Areas (DMAs), where the flow and the pressure at the inlet are monitored [3]. Recently, classifiers have been proposed to analyze the residuals in [4]. In this

paper, we focus on the distribution network of the WSS and, in particular, on the problem of leakage detection and localization. We consider three main technical solutions for addressing this problem: (i) the utilization of embedded devices to be deployed at the nodes and along the transportation pipes, for collecting demand measurements and acquiring vibration signals due to the water flows; (ii) a local form of intelligence, running at the gateway nodes collecting information about the vibration signals, for removing noise sources (such as the vibration signals caused by a vehicle on the road) not related to water leakages; (iii) an high-level aggregation scheme, responsible of mapping the data collected from the whole network or DMA (demand, pressure, presence of leakage) into a per-node estimate of presence/absence of leakages. For designing our scheme, we used a small-scale testbed, devised to optimize the local intelligence for filtering the vibration measurements, and a well known WSS simulation tool, called EPANET, for large-scale analysis. The simulator has been extended for emulating the innovative sensing system based on acoustic signals. Our solution has some interesting practical implications. First, it does not require a complete knowledge of the WSS topology and any calibration of an hydraulic model. Second, it only needs data from normal operation behavior. The remainder of the paper is organized as follows. An analysis of the state of the art can be found in Section 2. The design of the vibration sensor is presented in Section 3, together with the experimental results in the local testbed. In Section 4 we provide performance results on the considered large scale scenarios, and discuss the results of our simulations. Finally, some conclusions are drawn in Section 5.

Related work

A leakage detection technique using acoustic emissions is based on the principle that water flows due to losses passing through a perforation in the pipe create an acoustic signal [5]. When a leak occurs, acoustic sensors installed outside the pipe, track and detect the acoustic signal as it propagates along the pipeline. Acoustic signals are greatly influenced by the distance between the perforation and the sensor, as well as by the background noise from the environment and those produced by a pipe burst. Therefore, their utilization requires some signal processing techniques, which can be expensive in large-scale networks. An alternative approach for monitoring WDNs is the usage of Micro Electro-Mechanical System (MEMS). Indeed, MEMS are low-cost solutions, with a relatively small dimension and low weight, which make them suitable for identifying leakages by means of vibration signals. MEMS accelerometers can have a range of acceleration from 0.5g to 200g (being “g” the gravity acceleration) and a bandwidth spacing from 10 Hz to order of tens kHz, which make possible the analysis of most type of vibration signals in the WDN field. In some cases, sensing can exploit both acoustic and vibration signals [6]. In [7], a method to identify leaks is proposed for identifying leakages even when they are far from the sensing points (blind spots), while [8] proposes models of buried pipelines to estimate wave velocities. In our work, we propose a small footprint of the sensing devices, assuming that they can be installed inside a pipe during repairs with a no-dig technology; in perspective, we can envision a scenario where pipes are continuously monitored against water leaks. Apart from the sensing and noise filtering problem, several works focus on data aggregation for leak localization. For this problem, black-box models based on Machine Learning (ML) are becoming a dominant solution. Different approaches, such as Support Vector Machines (SVMs), k-NN classifiers or multi-layer neural networks have been investigated in [9] [10] [11], respectively. We also exploit ML for aggregating the data collected by the sensors deployed within the network.

Leakage detection using vibration signals and AI on board

In this section, we describe the design and implementation of our sensing element for water leakages, based on accelerometer measurements and on embedded processing. We chose a commercial device by STMicroelectronics, with a low noise MEMS-based accelerometer IIS3DWB, featuring a programmable full-scale measurement and set up to measure in a range of

$\pm 2g$ in each of the three axes. The system works with a sampling rate of 26.71 kHz, thus providing a new measurement every $37\mu s$. Acquired data are digitally converted with a 16-bit Analog to Digital Converter (ADC) integrated into the accelerometer, reaching a nominal resolution of about $61\mu g$. Since terrestrial gravity, affecting vertical direction, contains no information related to pipe vibrations, it's possible to remove the terrestrial acceleration component to maximize precision. The sensor is interfaced with a low power 32-bit microcontroller, namely STM32L4R9ZI by STMicroelectronics as well. The microcontroller is responsible of running the processing tasks for extracting some relevant features from a temporal sequence of measurements and classifying the sequence as a leakage/non-leakage event. Indeed, as discussed in [12] by Youngseok et al., the spectrum of the vibration signal in case of regular functioning of the pipe is quite different from the one acquired during the leakage: in presence of leakages, there are some frequential components in the range from 400 Hz to 5 kHz, according to the entity of the leakage and the dimension and material of the pipe. The amplitude of the sensed vibration varies based on the distance between the leakage and the sensor and it can change depending on the amount of wasted water. Since the vibrations are in the acoustic range, we designed a spectrum analysis scheme based on typical solutions for speech recognition: More into details, we extracted the Mel Frequency Cepstral Coefficients (MFCCs) [13] of different groups of measurements. We organized consecutive samples of the accelerometer measurements into temporal windows of 0.46 s (corresponding to 6 groups of 2048 measurement samples). The sequence is divided into 9 sub-windows of 2048 measurements, partially overlapping, and each window is converted in 40 MFCCs with a float precision.

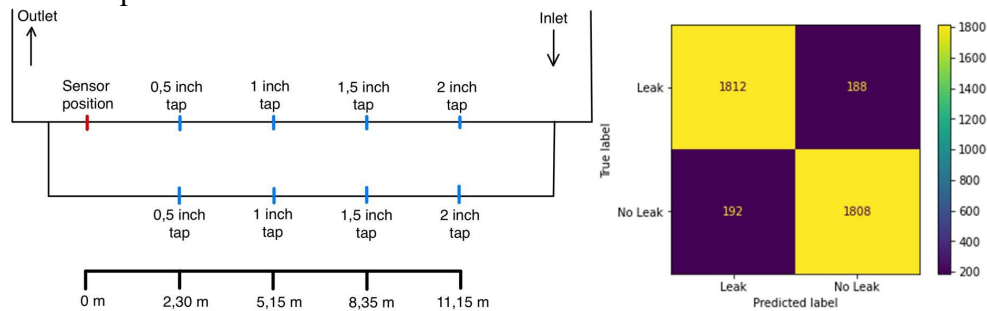


Figure 1 - Scheme of the used pipe system (left), and confusion matrix of the model performance (right).

The MFCCs matrix of 40x9 coefficients is then used as an input of a multilayer perceptron network, with two 256-neurons fully connected layers by a flattening layer. In order to classify the input as affected by a leakage, the output of the algorithm produces a number between 0 and 1 which represents the probability of a regular working condition of the sensed pipe. For the purpose of classification, outputs greater than or equal to 0.5 have been classified as "no leak" while outputs less than 0.5 are considered "leak".

Smart pipe performance evaluation in real testbed scenario

For training our model, we collected some experimental data by exploiting a small-scale testbed with real pipes and taps to be used as controlled leakages. The taps are different in diameter and distance from the sensor, as depicted in Figure 1(left). During data acquisition, the electronic system has been placed on the pipe and various scenarios, in terms of taps opened over time, have been recorded as a ground-truth, together with the relevant measurements' traces. The collected data has been split in training, validation and test set, and then used to train the model and evaluate its performances. The tested topology produced a model that is able to correctly classify samples extracted from a leaking condition 90.4% of the time and 90.6% of the time in the case of a regular functioning pipe, as shown in the confusion matrix represented in the rightmost part of Figure 1.

After the evaluation, the model has been tested on the complete recorded scenarios, even with unseen scenarios, showing promising results. The performance of the algorithm can be improved by introducing some memory effects: indeed, in case of leakages, it is very likely that the leakage will last for long time intervals and therefore consecutive outputs can be averaged for filtering sporadic classification errors.

Leakage detection in large scale scenario

For analyzing the possibility of identifying leakage events in real WDNs, we worked on simulation, where it is relatively easy to build a complex topology of a water grid, comprised of a number of links connected together to form loops or branches. Each link may contain one or more pipes (with different diameters) connected in series. A pipe is a segment of a link that has a constant flow with a constant diameter, and no branches. Two or more pipes are joined by node elements. Water can leave the network at these nodes, where we also assume that smart readings of water demand are possible. For simulating a realistic scenario, we made some assumptions on the sensors and meters available in the system, as well as on the communication networks deployed for transporting the data to a central aggregation server (working at different scales, including a district-level scale). These assumptions are used for filtering the whole trace produced by the simulator (which characterizes the complete view of the system) into a set of measurements visible to the leakage detection system. The system works on the basis of two ML models: a low level one, which only runs on the sensors for processing acceleration measurements and identifying leakage events related to the node where the sensor is deployed; a high-level one, which uses all the readings produced by smart meters and sensors and flow balancing conditions for identifying the overall leakages in the network (even where sensors are not available).

Communication Network

We assume that the system works thanks to IoT technologies. According to the WDN architecture, we can expect to adopt cellular technologies, or technologies working in unlicensed bands. For our analysis, we focused on the usage of LoRaWAN [14], a wireless technology offering low-cost, low-power, and long-range communications (up to a few kilometers). The architecture is based on a start of star deployment, where End Devices (EDs) transmit data to multiple nearby Gateways (GWs) deployed in their covered area, which in turns forwards frames to the Network Server (NS) and eventually to the corresponding IoT applications server. We assume that smart readers and sensing elements act as EDs, equipped with a LoRaWAN interface, and that at least one GW is able to collect the measurements provided by the EDs.

Water system simulator

Our work is based on the Water Network Tool for Resilience (WNTR) [15]. The tool is a Python package designed to simulate and analyze the resilience of water distribution networks based on EPANET, an open source software for modeling hydraulic and quality dynamics of a WDN) [16]. WNTR was developed to extend the capabilities of EPANET and simulate the dynamics of water flows across pipelines, taking into account bulk flows and pipe wall reactions, as well as the availability of water sources and reservoirs. WNTR has an application programming interface (API) that is flexible and allows the configuration of the network topology and the scheduling of disruptive incidents and recovery actions. We use the WNTR simulator to generate a suitable dataset to feed and test the ML approach. On one side, the simulator generates a complete trace, with the status of each node over time, by recording a log file containing the following fields: *timestamp*, *nodeID*, *demand*, *head (elevation + pressure head)*, *pressure*, *has_leak*, *smart_pipe*. On the other side, we filter a subset of data corresponding to smart meters and sensors, together with the real status of nodes, for training a ML model whose input is the total set of measurements in a given time window, and whose output is a vector with the status leakage/non-leakage of each

node. An interesting feature of our model is the lack of input data related to the network topology. Indeed, topological information are indirectly retrieved thanks to the balancing of water flows over time, in presence of time varying water demands. For testing our approach, we used some pre-defined network models offered by the Open Water Analytics's community public repository. Specifically, we considered three different sub-networks [17], named A, B, and C, with an increasing level of complexity (in terms of the total amount of nodes). At edge nodes, we also defined a set of demand-patterns, which represent water consumption at different timesteps. We also considered an interval of one hour as minimum time interval for changing the demand patterns. In one half of the nodes, we added the possibility of adding leakage events. We then run simulations for each network topology and for different demand-patterns, lasting a total duration of one week or one month. Pipes with leaks are chosen from a pseudo-random function in the simulation script (coded in Python) and feed into the entire simulation's time intervals.

Table 1 - Parameters of three considered network.

Network	Total network demand [gal/min]	Leak demand [%]	# of nodes	# of reservoirs	# of nodes with leakage	# of smart pipes
A	0.00333947	43.45	83	1	41	20
B	0.13837243	48.74	1038	1	518	259
C	0.30645519	42.38	2099	2	1048	524

Table 1 shows the list of the parameters present in each WDN network used to generate the dataset. For each network, we report the total amount of the demand from the nodes and the percentage of nodes involved in water leakages. The table also summarizes the total number of network nodes, the total number of reservoirs, the number of nodes with leakages and the number of installed smart pipes (equipped with the sensing element working on acceleration data). We represent our use case as a Supervised Classification problem in which we try to predict if there is a leakage on the pipe. From the simulation trace, we extracted our training set with the following features: *demand*, *head*, *pressure*, *smart_pipe*; this latter feature can assume three different values: 0, if the vibration sensor is present and a leakage is not detected; 1, if the vibration sensor is present and the leakage is detected; 2 when there is no sensor. Since this is a classification problem, our test set also contains the status of each node, in the boolean variable *has_leak*.

Performance results

In order to choose the most adequate classification model, we applied different techniques and confronted them. We tested the following classifiers: KNeighborsClassifier, LinearSVM, RBFSVM, DecisionTree, RandomForest, AdaBoostClassifier, GaussianNaiveBayes. Models have been compared in terms of accuracy on the node classification. We analyzed accuracy results for the trained model, in the three different network topologies (A, B and C), when only one week of data is available and without the usage of vibration sensors. Accuracy values are calculated after the fitting and prediction phases and after the confusion matrices generation. Specifically, we extract the *True Positives (TP)*, *True Negatives (TN)*, *False Positives (FP)* and *False Negatives (FN)* values and calculate the accuracy as seen here: $(TP + TN) / (TP + TN + FP + FN)$. Each classifier is evaluated through a series of $k=\{7,4\}$ cross-folds. The maximum accuracy (**about 70%**) is reached for **network A**, which has a lower number of nodes and therefore a simpler topology to be inferred. The presence of the vibration sensor in all the considered networks, improve the accuracy of the system about the **10%**. Finally, we analyze the impact of a longer data trace, corresponding to a whole month of data, for training the system equipped with the smart pipes. We note that longer traces are able to solve most of topological ambiguities and achieving very high accuracy (as high as **98%** for a model based on Decision Trees) even for the more complex network topologies B and C.

Conclusion

This work presented a system architecture and a data-driven approach to detect leakages in WDN. A LoRaWAN IoT network is considered to collect information at WDN nodes, where water demands are measured over time. Moreover, in a sub-set of nodes we also assume to deploy innovative sensors, equipped with embedded intelligence, for processing accelerometer measurements and identify leakage events. We demonstrate that demand measurements over time can be exploited for retrieving topological information of the network thanks to the balance conditions among flows. Starting from a realistic trace of water demand measurements and optional smart sensor readings, a model can be trained for identifying leakage events at each network node. Although our model has been trained and validated by exploiting numerical simulation, we expect that results can be easily generalized to real deployments.

Acknowledgments

This work has been partially financed by the Project “TiSento” (Azione 1.1.5. - POC Sicilia 2014/2020 Asse 1 - PO FESR 2014/2020).

References

- [1] WWAP: World Water Assessment Programme, The United Nations world water development report 2020: water and climate change: WWDR2020,978-92-3-100371-https://unesdoc.unesco.org/notice?id=p:usmarcdef_0000372985
- [2] Europe’s Water in Figures An overview of the European drinking water and waste water sectors 2021 edition EurEau ISBN 978-2-9602226-3-0.
- [3] Puust, R.; Kapelan, Z.; Savić, D.A.; Koppel, T. A review of methods for leakage management in pipe networks. *Urban Water J.* 2010, 7, 25–45. <https://doi.org/10.1080/15730621003610878>
- [4] Khulief, Y.; Khalifa, A.; Mansour, R.; Habib, M. Acoustic Detection of Leaks in Water Pipelines Using Measurements inside Pipe. *J. Pipeline Syst. Eng. Pract.* 2012, 3, 47–54. [https://doi.org/10.1061/\(ASCE\)PS.1949-1204.0000089](https://doi.org/10.1061/(ASCE)PS.1949-1204.0000089)
- [5] Adedeji, Kazeem & Hamam, Yskandar & Abe, B.T. & Abu-Mahfouz, Adnan. (2017). Towards Achieving a Reliable Leakage Detection and Localization Algorithm for Application in Water Piping Networks: An Overview. *IEEE Access.* 5. 20272 - 20285. <https://doi.org/10.1109/ACCESS.2017.2752802>
- [6] Fuchs, H.; Riehle, R. Ten years of experience with leak detection by acoustic signal analysis. *Appl. Acoust.* 1991, 33, 1–19. [https://doi.org/10.1016/0003-682X\(91\)90062-J](https://doi.org/10.1016/0003-682X(91)90062-J)
- [7] Yang, J.; Wen, Y.; Li, P. Leak location using blind system identification in water distribution pipeline. *J. Sound Vib.* 2008, 310, 134–148. <https://doi.org/10.1016/j.jsv.2007.07.067>
- [8] Muggleton, J.; Brennan, M.; Pinnington, R. Wavenumber prediction of waves in buried pipes for water leak detection. *J. Sound Vib.* 2002, 249, 939–954. <https://doi.org/10.1006/jsvi.2001.3881>
- [9] Ferrandez-Gamot, L.; Busson, P.; Blesa, J.; Tornil-Sin, S.; Puig, V.; Duviella, E.; Soldevila, A. Leak Localization in Water Distribution Networks using Pressure Residuals and Classifiers. *IFAC-PapersOnLine* 2015, 48, 220–225. <https://doi.org/10.1016/j.ifacol.2015.09.531>
- [10]. Wachla, D.; Przystalka, P.; Moczulski, W. A Method of Leakage Location in Water Distribution Networks using Artificial Neuro-Fuzzy System. *IFAC-PapersOnLine* 2015, 48, 1216–1223. <https://doi.org/10.1016/j.ifacol.2015.09.692>
- [11]. Verde, C.; Torres, L. *Modeling and Monitoring of Pipelines and Networks: Advanced Tools for Automatic Monitoring and Supervision of Pipelines*; Springer: Berlin, Germany, 2017. <https://doi.org/10.1007/978-3-319-55944-5>
- [12] Kim, Youngseok & Jung, Haewook & Ryou, Jaesuk & Choi, Jaehyuk. (2021). A Basic Experimental Study on Analysis of Leak Signal and Monitoring Method for Water Supply Pipe. *Applied Sciences.* 11. 2097. <https://doi.org/10.3390/app11052097>
- [13] Ittichaichareon, C., Suksri, S., & Yingthawornsuk, T. (2012, July). Speech recognition using MFCC. In *International conference on computer graphics, simulation and modeling (Vol. 9)*.
- [14] LoRa Alliance Technical Committee, “LoRaWAN 1.1 Spec.” <https://lora-alliance.org/resource-hub/lorawantm-specification-v11>, 2017.
- [15] Klise, K.A.; Bynum, M.; Moriarty, D.; Murray, R. A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study. *Environ. Model. Softw.* 2017. <https://doi.org/10.1016/j.envsoft.2017.06.022>
- [16] Jun, L.; Guoping, Y. Iterative Methodology of Pressure-Dependent Demand Based on EPANET for Pressure-Deficient Water Distribution Analysis. *J. Water Res. Plan. Manag.* 2013, 139, 34–44. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000227](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000227)
- [17] https://github.com/gabrielication/py_epanet2csv/tree/main/networks