Vibration-based water leakage detection system for public open data platforms

Giuseppe Costantino Giaconia[1], Fulvio Lo Valvo[1], Karim Ladjery[2], Francesco Di Puma[2], Josè Falla[2], Ioannis T. Christou[3], John Soldatos[3], Thanasis Papadakis[3], Boian Popunkiov[4], Daniela Alexieva[5] and Dian Raykov[5]

[1] Dep. of Engineering, Università degli Studi di Palermo, viale delle Scienze Palermo, Italy {costantino.giaconia, fulvio.lovalvo}@unipa.it
[2] EKSO srl, Contrada Tabuna, 97100 Ragusa RG, Italy {k.ladjeri, f.dipuma, j.falla@ekso.it}
[3] Dept. of Research & Innovation Development - Netcompany-Intrasoft, Luxembourg, Luxembourg {Ioannis.Christou, John.Soldatos, Athanasios.Papadakis}@netcompany.com
[4] Burgas Municipality – 26, Alexandrovska Street – Burgas, Bulgaria {b.popunkyov@burgas.bg }
[5] Innovative Systems-Burgas Ltd – Burgas, Bulgaria {integrirangradski, draikov}@abv.bg

Keywords: Vibration, MEMS, Leakage Detection, Smart Cities

Abstract: Water Distribution Networks are known to lose a consistent percentage of drinkable water due to the presence of leakages. In this paper it is proposed a solution to detect water leaks consisting of: i) a new sensing equipment able to acoustically monitor the external surface of a newly laid underground pipe; ii) a training of several machine learning models able to analyse the data collected by the new sensing equipment; iii) a user dashboard to give the final user the possibility to monitor the pipe's condition. The research process included the generation of artificial leakages capable to produce a suitable dataset necessary to properly train machine learning models onto.

1. Introduction

The issue of water leakage in public infrastructure has become increasingly prevalent in recent years. Aging pipes, which have been in service for decades, are beginning to show signs of wear and tear. This deterioration often leads to serious leakages, resulting in significant water loss. In many European countries, the situation is particularly dire. The amount of drinkable water lost due to poor pipe conditions exceeds 40% in some areas, and in certain regions, this figure reaches a 70%. This poses a serious challenge to the sustainability of water supply systems.

Water pipes, by their very nature, are often installed underground and can run for several km. This makes the task of detecting leakages particularly difficult. Traditional methods of leakage detection, such as visual inspection or listening devices, are often ineffective due to the inaccessibility of the pipes and usually can't perform continuous monitoring. Furthermore, these methods are labour-intensive and time-consuming, making them impractical for large-scale water supply networks.

This paper proposes a novel approach for leakage detection using vibration sensors. These sensors able to detect the unique vibration patterns produced by water leaks. By installing these sensors along the length of water pipes, it is possible to monitor the water distribution network for potential leaks.

However, the sheer volume of data generated by these sensors presents another challenge. It is impractical to manually analyse this data for signs of leakage. To address this, the proposed system employs Machine Learning (ML) models.

The use of ML models not only increases the efficiency of the leakage detection process but also improves its accuracy. Unlike human operators, ML models do not suffer from fatigue or loss of concentration. They can operate around the clock, ensuring that leaks are detected as soon as possible.

The problem of water leaks is being addressed on many levels (El-Zahab & Zayed, 2019), (Mohd Ismail, et al., 2019). There is an effort in developing solid architectures to perform continuous monitoring as suggested by (Aziz, et al., 2022). A recent literature review has categorized the methodologies for detecting and locating leaks into data-driven approaches and model-based methods (Nimiri, et al., 2023). Data-driven approaches demand efficient exploitation and use of available data from pressure and flow devices, while model-based methods require finely calibrated hydraulic models to reach a verdict.

Another review also highlighted the use of model-based and data-driven approaches for leak detection (Hu, et al., 2021). It noted that while model-based approaches require highly calibrated hydraulic models, their accuracies are sensitive to modeling and measurement uncertainties.

Moreover, a novel intelligent monitoring-warning system for leakage detection has been proposed (Li, et al., 2022). This system consists of a leakage locating model and a leakage quantity model, aimed to provide valuable insight into the construction and maintenance of future rural water supply projects.

These advancements in leakage detection technology, coupled with the use of Machine Learning models, are promising steps towards addressing the water leakage problem more effectively and efficiently.

2. Methods

The system was developed with the purpose of detecting leakages by monitoring the pipe vibrations, as shown by (Martini, et al., 2013). The waterflow produces vibrations that can be sensed with modern industrial Micro-Electro-Mechanical-Systems (MEMS) solutions. The data collected by the new system is sent to a server, making it available for machine learning and monitoring purposes.

To test the effectiveness of the system, a leakage scenario was simulated using a fire hydrant devoted to emulating a possible leakage with a predefined timing pattern. This allowed for the generation of real-world data, replicating conditions that the system would encounter in a practical setting.

The next phase involved the application of machine learning techniques to the collected data. The objective was to train a model that could accurately identify and predict leakages based on the vibration patterns. This involved training, testing cycles, and fine-tuning the models.

Finally, the results of the machine learning model were made accessible to the end-users through a graphical user interface showing a suitable dashboard.

3. Hardware

3.1 Sensing System

The pipe vibrations were acquired using a newly developed system consisting of 10 sensors subdivided in two chains. These chains were connected to a main computer to consistently transfer acquired data to a cloud server equipped with large storage space able to record up to 6 months of continuous datastream. Acquired data has then ben elaborated using ML algorithms as described in paragraph 4.

The experimental campaign took place in an urban area, where the sensing system was applied to a 135 Ø mm new underground water polyethylene pipe.

Sensors were positioned over the pipe's surface, using a low acoustic impedance glue, at distances between each other as displayed in Table 1. Distances are shown as absolute gaps starting from sensor 11, relative gaps from the previous sensor or absolute distances from the simulated leakage. A more comprehensive picture of sensor disposition is depicted in Figure 1 where continuous lines represent an electronic connection between the 2 nodes.

Sensor #	Distance (m)		
	From 11	From prev.	From leak
11	0	0	93,75
9	7,75	7,75	86,00
7	18,95	11,20	74,80
5	28,75	9,80	65,00
3	39,25	10,50	54,50
10	48,45	9,20	45,30
8	58,45	10,00	35,30
6	66,75	8,30	27,00
4	76,35	9,60	17,40
2	86,45	10,10	7,30

Table 1 Inter-sensor distances.

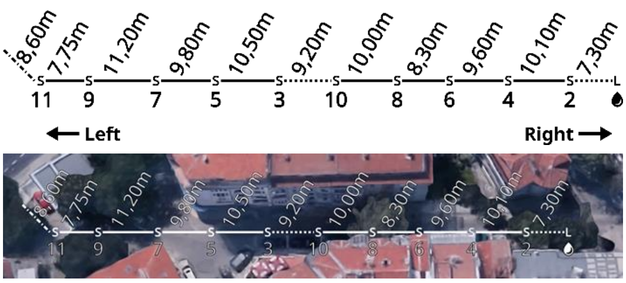

Figure 1 Sensor's location.

An average 10 meters distance between sensors has been chosen as a result of a previously carried out experimental campaign where simulated leaks were detected by similar sensors' prototypes.

To sense the small vibrations produced by waterflow in civil pipelines, the hardware needed to be designed around a sensing element able to capture that information.

Among the many accelerations sensors available, the selection focused on MEMS accelerometers, which are known for their small size, low costs, and their ability to sense over a relatively large bandwidth, starting from very low frequencies. This approach has proven to be promising in (Khulief, et al., 2012), (Mistretta, et al., 2018), (Restuccia, et al., 2023) and (Lo Valvo, et al., 2023).

The accelerometer selection spaced many manufacturers and models with different characteristics. Since water produces very low amplitude signals and within a relatively low range of frequencies, a very low noise and a high sampling frequency are required. Among the different sensors, the chosen one is the IIS3DWB (from ST Microelectronics) which is able to acquire vibration data at a high speed. In the experimentation, this sensor was set to use a scale of ±2g where g is the gravity acceleration. This accelerometer also features a very high sensitivity, reaching a noise level of about $75\frac{\mu g}{\sqrt{Hz}}$. Summary of its main interesting features are given in Table 2.
The IIS3DWB sensor communicates using SPI, a low range, high speed serial protocol (STMicroelectronics, 2020).

Sensor features	
Number of axes	3
Full-scale	±2/±4/±8/±16 g
Response range	from dc to 6 kHz
Noise density	$75\frac{\mu g}{\sqrt{Hz}}$ in 3-axis mode
Sampling resolution	16 bit per axis

Table 2 Accelerometer features

3.2 Microcontroller

Since the accelerometer is capable of communicating on short distances, a microcontroller is required to be able to install different sensors at a distance.

The STM32L476RG (STMicroelectronics, 2019) is a low power microcontroller equipped with an ARM Cortex-M4 processor unit, 1MB of Flash memory and 128 KB of RAM.

The microcontroller is responsible for the acceleration data acquisition, a partial elaboration, and the communication of the data to the single board PC.

The final PCB design is 85x28 mm in size and includes all the electronics needed for the correct behaviour of the sensor. A 3D representation of the device is shown in Figure 2.

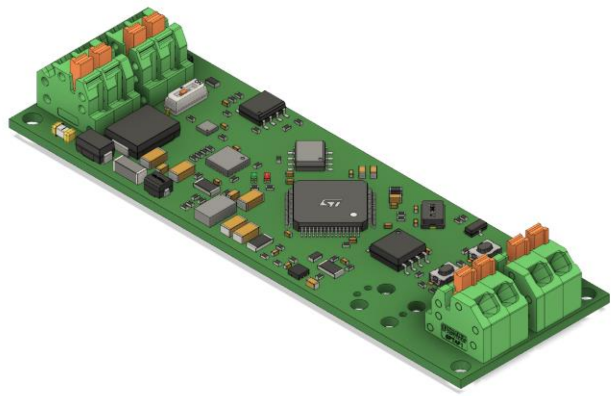

Figure 2 3D view of the final device.

3.3 Serial Communication Protocol

The communication protocol for the two chains needed to:

- reliably cover long distances
- have a simple connection scheme
- support high speed rates

The final choice for the bus communication ended up being the RS485, which respected the requirements. Since a lot of data is produced, even the RS485 wouldn't be enough to transfer data coming from 10 sensors, they have been split in two chains (odd and even) connected to the same sink node. With this approach, the system is capable to deliver acceleration samples at a rate up to 1700 samples per seconds (sps). Due to the described conditions, a sample speed of 1600 sps have been setup taking also into account that each accelerometer hasn't exactly the same output data rate (ODR). However this approach grants the ability to detect frequency in the range of 0÷800 Hz, which comprehends most of the necessary informations.

3.4 Single board computer

The transferring of data to the internet was done using a Raspberry Pi (RPi), a small, single board computing element with enough computing power to perform data polling, compression, and delivery to the cloud.

The RPi is also able to upgrade the firmware installed on the devices via Over The Air (OTA) mechanism.

3.5 Installation

Before the on-site installation, the sensors were glued to waterproof boxes, previously drilled for sealed cable allocation. A curved surface was added underneath the box to make better acoustic contact with the outer surface pipe.

The installation happened concurrently to the process of pipe layout. When the pipe was already inside the trench, the sensors were glued to the pipe and cabled together, as shown in Figure 3.

4. Software

4.1 Machine Learning

The original data gathered in the server are high-frequency data recording the vibrations measured by the installed sensors on the pipe. As such, they record the fluctuations of the amplitude of the vibrations measured, whether they are caused by water waves hitting the surfaces of the water pipe, but also due to surface vibrations caused for example by heavy vehicles moving in the vicinity. As a result, the sensors can often record very high values that are not due to leakage, but instead have external causes.

Figure 3 Example of sensor with enclosure glued to the pipe.

As shown in Figure 1, the experimental data used for this study isolated only a single leakage located to the right of each sensor. Given the uniformity of the sensors used in this experiment, an individual machine learning model was constructed for each sensor, taking into account the distances of each sensor from the leakage source. For instance, sensor #2's model determines if there is a leakage 7,3 meters to its right, while sensor #4's model identifies a leakage 17,4 meters to its right, and so forth.

For each sensor, two features were derived for both detection and localization of the leakage: the first feature aggregates the values read by the sensor in non-overlapping intervals of 0.1 seconds; in particular, the original data come in tuples of (<timestamp>, <value>) pairs, we take the *average avg* of all values that lie in the time-span [0, 0.1), and we create the new aggregate tuple (0.1, avg_1) then for values in the interval [0.1, 0.2) we create the aggregate tuple [0.2, avg_2) and so on. The second feature maintains the highest recorded value in the previous 10 seconds. This pre-processed data set (derived from the original raw data in a single scan) results in 10 models (one for each sensor) whose accuracy is maximum for the closest sensor (#2) with a test-value of 98,5% and is minimum for the most distant sensor (#11) with a test value of 97%. Each of these 10 models corresponds to a set of rules produced by the RIPPER-k (Cohen, 1995) rule extraction algorithm as implemented in the WEKA machine learning environment (Hall, et al., 2008).
The RIPPER-k algorithm is a rule-learning system that works by first separating the training dataset into a grow-set and a prune-set, and then starting to create conjunctive rules for the minority class, starting with an empty rule, and then sequentially adding clauses of the form "attribute=value" for categorical attributes, and "attribute<=value", "attribute>=value" or "attribute=value" for numerical attributes, until the rule is 100% correct on the grow-set (i.e. there are no data belonging to the majority class that satisfy all the clauses of the rule) until the entire minority class is "covered" by one or more rules. The choice for which attribute-value pair should be added to the rule next is mandated by the FOIL information gain criterion, see (Quinlan, 1990) similar to the choice of what attribute to branch on decision trees. As soon as a rule is constructed, it is pruned by iteratively deleting the clause from the rule that maximizes a particular function on the prune-set, until no deletion further improves the value of this function. The final ruleset, after a series of

optimizations forms the classifier that is used as follows: if a test instance satisfies all conditions of any rule in the rule-set, the instance is classified as belonging to the minority class, otherwise the instance is classified as belonging to the majority class.

Regarding localization, we design a "thought experiment" to see if we can detect leakages located in locations other than the one we have data for, further to the right of the location of the leakage in the experimental data. This should be possible, given that each sensor Y, using the data in the previous experiment, in reality learnt a model that answers the question Q(Y,L)="Is there a leakage L meters to my right?". Therefore, if a leakage is located approximately at a multiple *k* of 10m. (approximately the spacing between sensors) to the right of the original leakage, if we "shift" the models of the sensors by k steps to the right, and we apply them to the new data of the sensors in these positions should answer the model question Q with a "Yes".

This new "thought-experiment" assumed a leakage located 20 m. to the right of sensor #2, in other words, 10m to the right of the original leakage (see Figure 1). We make the assumption that the data that were in the original experiment read by sensor #11 (who is 93.75m away from the real leakage location), were now read instead by sensor #9 (who is in reality 86m away from the real leakage location); then the data that were really read by sensor #9 were read by sensor #7 (who is in reality 74.8m to the left of the real leakage location) and so on until we reach the right-most sensor (#2). For sensor #11 (the left-most sensor), we assumed that its data corresponded to a set of measurements that are read when there is no leakage; and we discard the data that sensor #2 read in the original experiment. With this arrangement, we expect that the model for sensor #11, when applied to the data (that was in reality read by sensor #9) will reply with "Yes". Since this answer means "there is a leakage 93.75m. to my right" this means that there is an indication that there is a leakage 93.75m to the right of sensor #9, and since sensor #9 is located 86m from the original leakage location, this means we have an indication for a leakage 93.75m - 86m = 7.75m (~10m) to the right of the original leakage, as we would like! Similarly, for each of the other models and data.
In general, by applying the 10 models we have already derived from the 10 individual sensors to each of the 10 datasets we have, we get a total of 10x10=100 decisions in one of the two possible forms:

A. "leakage answer indidates that sensor_X_detects_leakage_N_meters_to_my_right", or
B. "no_leakage"

The decision making in case (A) is as follows: assume that sensor X is located M m. to the left of sensor Y (M can be negative), and sensor Y is located L m. to the left of the (original) leakage. If the model trained for sensor Y data when applied to data read by sensor X says "leakage", then the decision is that "sensor_X_detects_leakage_N_meters_to_my_right" where N=M+L.

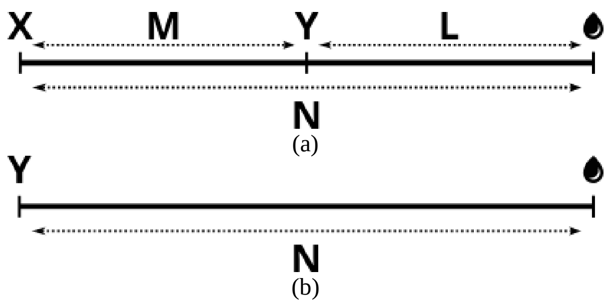

Figure 4 The model trained with data coming from sensor X (a) can be used on sensor Y to find leaks N meters away (b).

This way, every sensor location provides 10 answers (1 answer for each of the 10 models applied to the data it reads), and ideally, one answer will be of the form (A) and the other 9 in the form (B). The average of the positions dictated by all answers of the form (A) give us the best estimate of the location of the leakage.

4.2 User Dashboard

The policy dashboard is essentially the middle communication layer between the user and the AI model. The present dashboard is giving the user the option to upload the available data. The data is then sent to the AI model which, after the correct preprocessing, is able to perform its predictions.

The current visualisation shows a line chart which is developed from the various vibration values and the timestamp of measurement showing the range of vibrations for each sensor at the given time. Furthermore, the data from the AI model is shown in green whenever the model predicts that no leakage is present and in red if the model has detected a leakage. Overall, the dashboard gives the user the opportunity to visually identify any leakages present in the pipe and to take the appropriate measures in order to solve the problem. Two sample dashboard outputs are shown in Figure 5 and Figure 6; one with a leakage and one without. It is possible to notice that in the trace without leakage present (Figure 6) there are some false positives. This behaviour does not undermine the regular operation of the system as the final user can take action considering the majority of the output results and thus only when a clear red signal is persistent onto the dashboard.

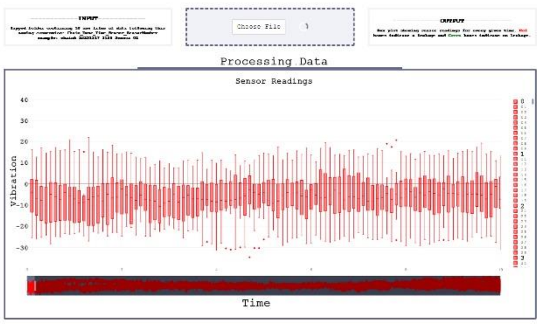

Figure 5 Dashboard when a leakage was simulated.

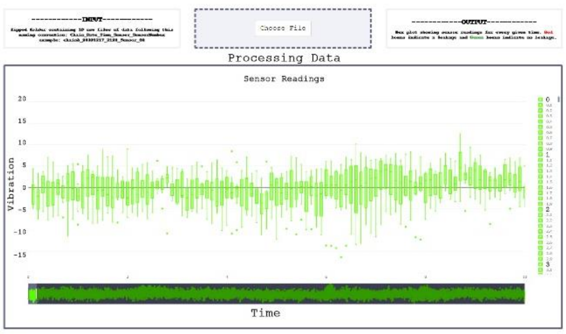

Figure 6 Dashboard when a leakage wasn't simulated.

These dashboards currently visualize the data from all 10 sensors combined. In future work it would be possible to expand software features adding the capabilty to show data from each sensor separately, allowing the user to select the data from a specific sensor to be displayed.

This will have two potential benefits. On one hand, it would deploy the refinements of the model, described in Section 5.1. On the other hand, it would also allow for more sensors to be deployed at various locations, without necessarily being connected in the same set up as this one. Furthermore, this would allow the possibility of having one common interface for the entire region or city, overlaid with a map, showing the location of each sensor.
Due to the limitations of this project timeframe, users must manually upload data corresponding to a 10-minute interval into the dashboard in order to display the graphs above. With the help of APIs this could be done automatically, allowing for continuous data monitoring and the issuance of an alert whenever a leakage is detected by the AI model. When the dashboard will be combined with an area map as described above, it will even give an approximate location of the leakage.

5. Results

For each dataset, we consider two variants, the first comprising only the aggregated sensor values in non-overlapping intervals of 0.1 seconds, and the second having in addition an extra value that contains the largest value read in the previous 10 seconds. From the results, it is obvious that when the aggregated data is not augmented with “memory values”, the classification accuracy is comparable to the toss of a coin. This is due to the fact that any value that is the simple aggregation of vibration amplitudes in an interval of 0.1 seconds is almost equally likely to appear in both possible scenarios of “leakage” or “no leakage”. In fact, by looking at the raw data, one sees that very often, in an interval of 0.1 seconds, the amplitude of vibrations recorded by any sensor can often reach values whose absolute value exceeds 300, but appear with both positive and negative sign, so that the average can be in the order of 0 or 10; other complicating factors are heavy vehicles travelling across the road forcing the sensors to record very high vibrations without the presence of a leakage. On the other hand, when the dataset is augmented with “memory values” that maintain the largest value read in an immediate previous short time interval, the accuracy of all ML models increases dramatically and becomes usable in a real production setting. The three models tested include (a) RIPPER, a rule-learner briefly described in section 4.1 (Cohen, 1995), (b) SVM, an implementation of Support Vector Machines (Vapnik, 1995) that are algorithms that attempt to find the hyperplane that best separates the two classes, leaving as much margin as possible from any point belonging to any class, and (c) ANN, an implementation of a one-hidden layer Neural Network with sigmoid nodes (Goodfellow, et al., 2016), that are networks of so-called perceptrons, “artificial neurons” inspired from the mammalian brain organisation, that combine their inputs via a dot-product operation with a set of weights and biases attached to each one, and then applying an activation function such as a sigmoid to this dot-product before sending it as output to the next layer of perceptrons. We chose these classifier algorithms as they are widely considered state-of-the-art in their specific domains, i.e. logic induction, wide-margin classifiers, and of course, neural networks and deep learning.

In Table 3, we report on the results regarding accuracy from several ML algorithms on a number of different datasets: (a) data coming only from the sensor closest to the leakage (sensor #2), (b) data coming only from the sensor furthest away from the leakage (sensor #11), and finally (c) combined data coming from all installed sensors. When one compares the results from the same classifier on the augmented datasets, it is clear that in all cases, the sensor closest to the leakage provides better accuracy to the sensor furthest from the leakage.

Finally, regarding the difference in performance of the three classifier algorithms (RIPPER-k, SVM, and ANN), the so-called No-Free-Lunch theorem (Wolpert & Macready, 1997) states that when averaged over all possible datasets the performance of any two classifiers is the same. However, in practice, this cannot be expected to hold, and in certain domains, one algorithm can obtain widely different performance than another. The high-performance accuracy of the rule-extraction algorithm probably implies that the feature space of this problem is likely partitioned by the two classes of the problem in hyper-parallelepipeds which can be easily found by rule-extraction systems but not so easily by neural networks or support vector machines.
When combining the 10 sensors’ features together (for a dataset having in total 20 features plus the target class), the same rule-extraction algorithm (RIPPER-k) provides overall test accuracy on unseen test data of 99.88%. This test accuracy refers to leakage detection using only a single instance of pre-processed data, i.e. data corresponding to only 10 seconds of raw sensor readings. Given that leakages cannot cease without human intervention, a second “test of time” using contiguous data (e.g. 5 minutes’ worth of data) to derive 30 consecutive “leakage/no-leakage” decisions and take the majority vote of those 30 decisions should be sufficient to remove any “false alarms” that could possibly be raised by test data (such as passing by heavy trucks.)

Sensor	Method	Accuracy (%)
2 (closest) no memory	RIPPER	57,24
	SVM	55,88
	ANN	55,27
2 (closest) 10s memory	RIPPER	98,22
	SVM	96,33
	ANN	96,43
11 (farthest) no memory	RIPPER	59,60
	SVM	55,88
	ANN	54,60
11 (farthest) 10s memory	RIPPER	97,96
	SVM	55,88
	ANN	56,06
All no memory	RIPPER	77,64
	SVM	58,66
	ANN	75,00
All 10s memory	RIPPER	99,88
	SVM	96,74
	ANN	98,28

Table 3: ML Models Accuracy on Leakage Detection

6. Conclusions

The proposed solution combines vibration sensors, machine learning algorithms and a dashboard useful for water leakage detection in underground pipes. The system was tested in an urban area, showing high accuracies in leakage detection.

This solution is competitive in terms of cost, continuous monitoring, and data accessibility. Nonetheless, it can be improved by adding more sensors to cover a longer distance, verify the robustness of the method for larger pipe diameters and better integrate the data with a local area map. A denser distribution of sensors network may also improve reliability of the overall monitoring system by compensating the possible malfunctions of single sensors, even if a thorough approach to increase resiliency of the proposed solution should be deeper studied in future work.

7. Acknowledgement

This study was partially funded by H2020, DT-GOVERNANCE-12-2019-2020 – Pilot on using the European cloud infrastructure for public administrations (Innovation Action), grant n. 101004480, AI4PublicPolicy and by the Project “TiSento” (Azione 1.1.5. - POC Sicilia 2014/2020 Asse 1 - PO FESR 2014/2020).
We kindly thank Dr. G. Greco and R. Rizzo, from University of Palermo, for their valuable support into the design and implementation of sensor systems.

8. References

Aziz, N. et al., 2022. Smart Water Network Monitoring: a Case Study At Universiti Teknologi Malaysia. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* 10 01, Volume XLVI-4/W3-2021, pp. 3-7.
Cohen, W. W., 1995. Fast Effective Rule Induction. *Machine Learning Proceedings 1995,* pp. 115-123.

El-Zahab, S. & Zayed, T., 2019. Leak detection in water distribution networks: an introductory overview. *Smart Water,* 6.Volume 4.

Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep learning.* s.l.:MIT press.

Hall, M. et al., 2008. The WEKA data mining software: An update. *SIGKDD Explor. Newsl.,* 11, Volume 11, pp. 10-18.

Hu, Z. et al., 2021. Review of model-based and data-driven approaches for leak detection and location in water distribution systems. *Water Supply,* 05 04.Volume 21.

Khulief, Y., Khalifa, A., Ben-Mansour, R. & Habib, M. A., 2012. Acoustic Detection of Leaks in Water Pipelines Using Measurements inside Pipe. *ASCE Journal of Pipeline Systems Engineering and Practice,* 5, Volume 3, p. 47.

Li, X. et al., 2022. A Novel Intelligent Leakage Monitoring-Warning System for Sustainable Rural Drinking Water Supply. *Sustainability,* 17 05.Volume 14.

Lo Valvo, F., Baiamonte, G. & Giaconia, C., 2023. Microcontroller Based Edge Computing for Pipe Leakage Detection. 4.pp. 16-22.

Martini, A., Troncossi, M. & Rivola, A., 2013. *Vibration Monitoring as a Tool for Leak Detection in Water Distribution Networks.* s.l.:s.n.

Mistretta, L. et al., 2018. *Embedding Monitoring Systems for Cured-In-Place Pipes.* s.l.:s.n.

Mohd Ismail, M. I. et al., 2019. A Review of Vibration Detection Methods Using Accelerometer Sensors for Water Pipeline Leakage. *IEEE Access,* Volume 7, pp. 51965-51981.

Nimiri, W. et al., 2023. Data-driven approaches and model-based methods for detecting and locating leaks in water distribution systems: a literature review. *Neural Computing and Applications,* 06 04, Volume 35, pp. 1-13.

Quinlan, J. R., 1990. Learning logical definitions from relations. *Machine Learning,* 5(3), pp. 239-266.

Restuccia, G. et al., 2023. *A distributed analysis of vibration signals for leakage detection in Water Distribution Networks.* s.l.:s.n.

STMicroelectronics, 2019. STM32L476 Datasheet. 4.Issue DS10198.

STMicroelectronics, 2020. IIS3DWB Datasheet. 4.Issue DS12569.

Vapnik, V. N., 1995. *The nature of statistical learning theory.* s.l.:Springer-Verlag New York, Inc..

Wolpert, D. H. & Macready, W. G., 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation,* 1(1), pp. 67-82.