Received: 14 August 2023 | Revised: 12 December 2023 | Accepted: 14 December 2023 | Published online: 27 December 2023

RESEARCH ARTICLE

Artificial Intelligence and Applications
2024, Vol. 2(2) 86–96
DOI: 10.47852/bonviewAIA32021513
BON VIEW PUBLISHING

Toward AI-Based Condition Monitoring and Predictive Maintenance for Water Smart Pipes: The SANDMAN Approach

Yacine Rebahi[1,*], Benjamin Hilliger[1], Patrick Lowin[1], Bowen Zheng[1], Giorgio Da Bormida[2] and Karim Ladjeri[2]

[1] *Fraunhofer Fokus, Germany*

[2] *EKSO S.R.L, Italy*

Abstract: Pipes age and corrosion are the main factors of leakage in water distribution networks. According to the World Resources Institute, European countries will face water problems by 2040. If we take Italy as an example, more than 40% of drinking water was lost in 2020 due to leaky aqueducts. Decrepit pipes can lead to environmental concerns, economical losses, and potential public health problems if water gets contaminated. Localizing leakage positions in an accurate way is often a big challenge. On the other side, replacing decrepit pipes is not an easy task and usually costly. An optimal solution to deal with water leakage is to use smart pipes where appropriate sensors monitoring the conditions of the pipes are incorporated in. Digitalization plays a crucial role here. By providing accurate information about the pipes and using artificial intelligence techniques for data analysis, potential leakages and their corresponding positions can be detected in time, which allows to schedule a maintenance task as soon as possible. The current paper discusses the use of smart pipes combined with predictive maintenance and shows how this combination improves water leakage detection, hence minimizing water waste and protecting the environment. The solution was validated in an experimental setup put in place by the Italian company EKSO S.R.L in its factory facilities in Rozallo, Italy. The obtained results show the feasibility of the solution and the relevance of using artificial intelligence techniques for predicting degradation in smart pipes.

Keywords: smart pipes, predictive maintenance, artificial intelligence, deep learning, LSTM, circular economy, Industry 4.0, SANDMAN project, AIREGIO project

1. Introduction

Water distribution networks are prone to leakage which is in general due to pipes age and corrosion (Alawadhi & Tartakovsky, 2020). According to the American Society of Civil Engineers (2021) and Gebelhoff (2023), there was a water main break every two min and roughly 6 billion gallons of treated water were lost every day in the United States (US). The situation in Europe is not much better. According to the World Resources Institute, European countries will face water problems by 2040. The World Wildlife Fund states that the water issues will affect 17% of the European people and 13% of Europe's GDP by 2050 (Karimli, 2023). If we take Italy as an example, it seems this country is wasting more and more water from leaky aqueducts. In a report published by the national statistics bureau "ISTAT [1]," it is stated that in 2020, Italy's aqueducts had lost 42.2% of the water they carried (Cinelli, 2023). Decrepit pipes can lead to environmental concerns, economical losses, and potential public health problems if water gets contaminated (Alawadhi & Tartakovsky, 2020). What makes the situation worse is the fact that water utilities are often not required to track water losses (Gebelhoff, 2023). Even if this information is provided, it is difficult to localize more accurately the leakage positions. On the other side, replacing decrepit pipes is not an easy task and often costly. An optimal solution to deal with water leakage is to use smart pipes (e.g., Figure 1) where appropriate sensors are incorporated in. In addition to the sensors, some communication means are also put in place which allows to monitor the conditions of the pipes and their contents. Although smart pipes are not widely deployed, the related technologies seem to be promising. A concrete example is given by the Italian company EKSO S.R.L[2] that has already experienced manufacturing pipes with embedded grids of Fiber Bragg Grating (FBG) from the production stage. FBG is a type of distributed net that reflects particular wavelengths of light and transmits all others. By combining signals coming from the pipe "sensors" set along the pipe, the pipes continuity/integrity can be evaluated. This is achieved by creating a periodic variation in the refractive index of the fiber core, which generates a wavelength-specific dielectric mirror. Hence, FBC can be used to block certain wavelengths or as wavelength-specific reflector.

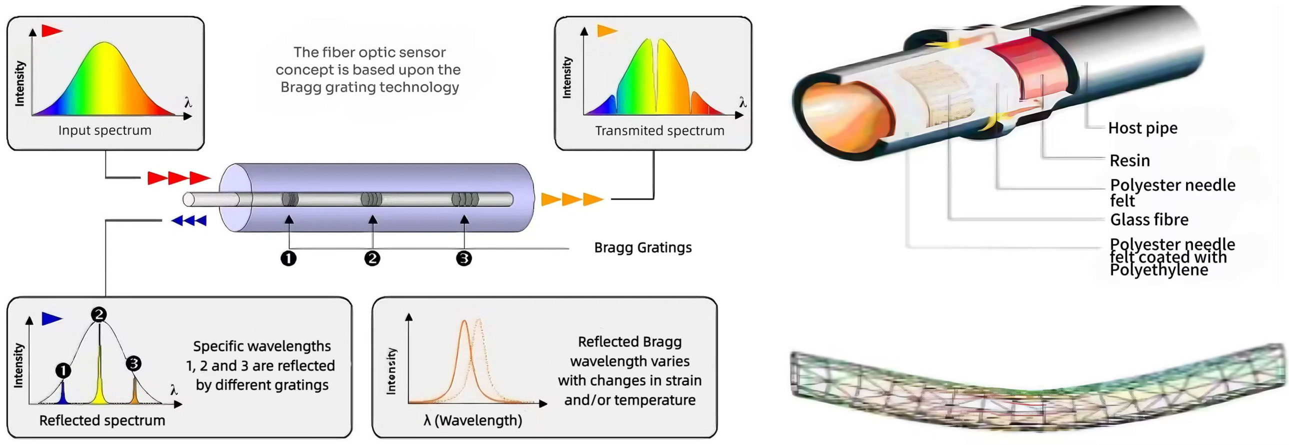
The main advantage of using smart pipes is the ability to monitor in a continuous way the status of the pipes and predict potential leakages in an accurate way and in time. Indeed, the data transmitted from the sensors will be analyzed using artificial intelligence techniques, and pipes conditions information will be made available to the water utilities operators. Combining the use of smart pipes with a predictive maintenance service has tangible results. In case, some

[1] https://www.istat.it/en/

[2] https://www.ekso.it/

*Corresponding author: Yacine Rebahi, Fraunhofer Fokus, Germany. Email: Yacine.rebahi@fokus.fraunhofer.de

Figure 1
(Left) FBG sensing approach. (Right) Smart pipe structure and digital twin initial approach



anomalies or potential leakages are accurately detected, the operators can schedule early enough a reparation task or a pipe replacement.

In this paper, we will discuss a pilot showing how the use of smart pipes combined with predictive maintenance can improve water leakage detection, hence saving vital nature resources. The pilot was achieved in the EKSO S.R.L factory facilities in Rozallo (RG), Italy (Figure 2). This location is the main EKSO production site, while it features more than 10 km of different types of smart pipes of many different cross sections and configuration just for experimentation purposes.

This paper is organized as follows. Section 1 introduces the topic being investigated. Section 2 discusses the related state of the art. Section 3 highlights the algorithm that was implemented, Section 4 presents the experimental results, and Section 5 concludes the paper.

2. Literature Review

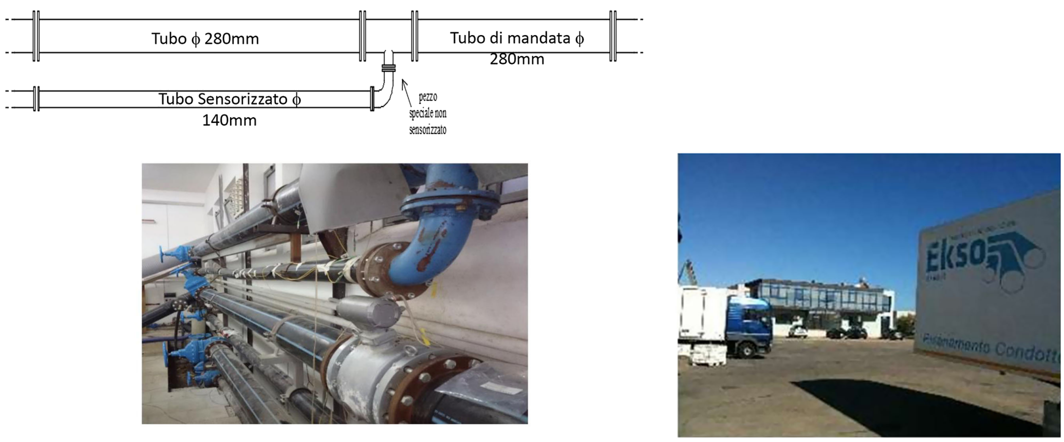
Exploring leakage detection in water pipes networks was tackled in the literature from different angles.

In Sadeghioon et al. (2014), a novel pressure sensing method able to measure pressure changes due to leaks was explored and its performance was shown in laboratory and field trials. These pressure changes can be used in detecting the corresponding leakage positions.

In Alawadhi and Tartakovsky (2020), it was shown that Bayesian data assimilation, combined with the method of distributions, can be a powerful tool for detecting small leaks in the presence of uncertain conditions and ambient noise. This approach was explored by the authors because it seems that describing transient flow in pipe networks using water hammer equations is not effective as factors such as initial and boundary conditions, and location and strength of a possible leak can render deterministic predictions of this system unreliable.

In Virk et al. (2020), detecting leaks and classifying their corresponding sizes were explored in wall-mounted pressurized water pipelines through vibrations measurements using low-power accelerometers. Here, three techniques, support vector machine (SVM), k-nearest neighbors, and decision tree, were used and their results were compared. Their simulation is based on MATLAB and shows that SVM provides the best results. This paper also provides a good survey of pipeline testbeds and their contributions to leak detection and localization. The approach followed in Virk et al. (2020) and the approach discussed in the current paper are similar; however, our approach explores the use of deep learning networks and vibration sensors in detecting leakages and their corresponding positions. Our implementation, which is meant to be part of a predictive maintenance service, is in Python, built on top of solid machine learning libraries such as Tensorflow, and uses TPUs to accelerate machine learning workloads.

Figure 2
EKSO pilot location in Rozallo, Italy



Maintenance is a service that technicians need to perform regularly or on demand to keep machines and equipment operational. Traditionally, maintenance can be seen as two categories, corrective or preventive. Corrective maintenance is applied when a given part is broken and needs to be repaired. Unfortunately, this approach might have long downtimes. For the preventive maintenance, the part will be replaced even it is not broken. This will be achieved, for example, by defining a fixed lifetime for each part. This approach can be cost inefficient because the replaced parts can still be in good condition and used for longer (Mobley, 2002; Rebahi et al., 2023). As maintenance is a cost driver in many industries with clear benefits (Lowin & Mihale-Wilson, 2021), data-driven predictive maintenance seems to be the optimal solution. This new approach aims at detecting machine failures, degraded performance, or a downtrend in product quality before one of these occurs (Lughofer & Sayed-Mouchaweh, 2019). Predictive maintenance is feasible nowadays, thanks to the advances related to internet connectivity, IoT platforms, cloud computing, and data analytics (Rebahi et al., 2023). It is based on analyzing the data collected by the sensors, and building models that can learn the machine behavior using past data from the machine (Zhu et al., 2019; Rebahi et al., 2023). As the amount of data generated by the sensors is in general huge, utilizing artificial intelligence and machine learning techniques appears to be conclusive and mandatory. As an example, in Gorenstein and Kalech (2022), several algorithms and AI techniques are proposed for economical replacement purposes. The authors look in particular at the adjacency of the components as replacing adjacent components could be efficient. The evaluation of the proposed solution was achieved on a real-world water transmission network. In Almobarek et al. (2023), a methodological framework for a predictive maintenance program for commercial buildings is proposed. The solution is developed for chilled water systems (CWS) and includes three parts, the setup, machine learning, and quality control. The results of the implemented framework seem to be encouraging as the accuracy of the prediction model is more than 98% for each CWS component.

It is worth to mention that the use of supervised or unsupervised learning depends on the availability of data labels reflecting related previous occurred failures information. If preventive maintenance is frequently applied, it would be very difficult to obtain such labels. For this reason, unsupervised learning seems to be a more reasonable approach to tackle data-driven predictive maintenance (Hilliger et al., 2023). As the current paper explores predictive maintenance in controlled environments, it is possible to generate data labels, and therefore the use of supervised learning appears to be the optimal option.

As anomaly detection is at the core of predictive maintenance with the focus on detecting anomalies in machines and equipment, Kamat and Sugandhi (2020) provided a survey describing the challenges related to the traditional strategies in this field. They also proposed a novel deep learning technique to detect a priori such anomalies. Another systematic literature review of machine learning methods applied to predictive maintenance was conducted by Carvalho et al. (2019). In their chapter, Sohaib et al. (2021) provide an overview of the deep learning algorithms utilized in predictive maintenance. Quality prediction was also addressed in the literature. Examples of applications range from quality predictions during production using sensor data to automated quality inspection in the field using measurement data. A comprehensive and systematic review of related scientific publications between 2012 and 2021 was conducted by Tercan and Meisen (2022). On the other side, activities covering condition-based maintenance were undertaken for instance by Sharma et al. (2022). After reviewing the related work in this field, they noticed that explainable artificial intelligence can help in providing unique insights and opportunities for addressing critical difficulties in maintenance, and leading to more informed decisions. With other respects, Hilliger et al. (2023) and Cardoso et al. (2023) discussed in detail some examples of long short-term memory (LSTM) techniques used for detecting anomalies in machines and how these techniques can be part of a global predictive maintenance solution.

3. Predictive Maintenance for Smart Pipes: The SANDMAN Solution

The predictive maintenance algorithm that we have developed has two main goals. The first goal is to predict whether a pipe is leaky, using vibration sensor data. The second goal is to classify the location and size of a leak, once it was detected. These goals were achieved using very similar techniques, namely extensive preprocessing and deep learning LSTM networks.

3.1. The LSTM approach

LSTM networks, introduced by Hochreiter and Schmidhuber (1997), are a type of recurrent neural network (RNN) architecture specifically designed to capture and model sequences of data, making them particularly effective for tasks involving time-series data, natural language processing, and speech recognition. LSTMs address some of the limitations of traditional RNNs, such as the vanishing gradient problem, which can hinder the ability of the network to learn long-range dependencies in sequences.

At its core, an LSTM unit is composed of four main components: a cell state, an input gate, an output gate, and a forget gate. These components work together to allow LSTMs to remember and forget information over varying time scales, making them capable of maintaining relevant context information even across long sequences.

To be more concrete, it is beneficial to look at each of these cells in more detail. An overview of the LSTM algorithm is depicted in Figure 3.

Cell State: The cell state serves as the long-term memory of the LSTM unit. It runs through the sequence and is updated at each time step. Information can be added or removed from the cell state using the input gate, output gate, and forget gate.

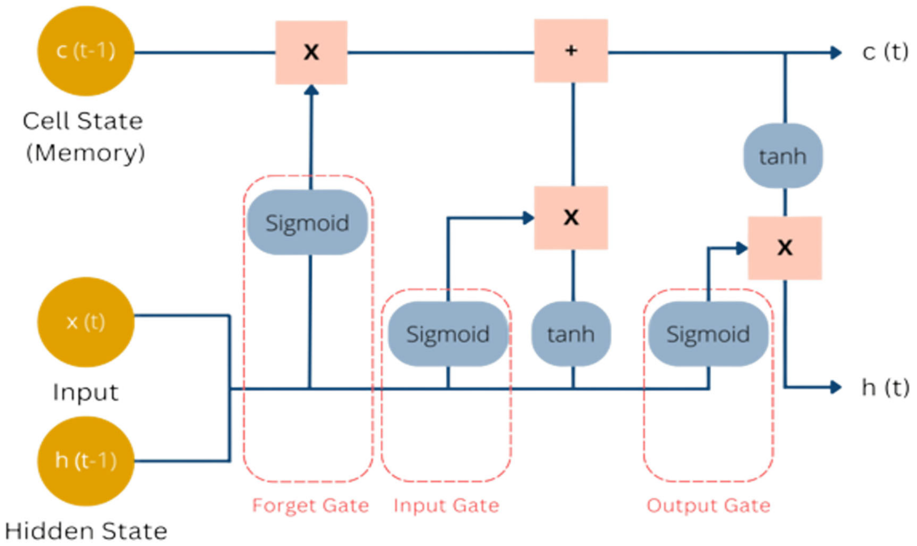
Input Gate: The input gate decides which information from the current input and the previous hidden state should be added to the cell state. It calculates a candidate update that can be added to the cell state and determines which parts of the candidate update should be included based on its sigmoid activation.

Forget Gate: The forget gate determines which information from the previous cell state should be retained and which should be discarded. It decides what information is irrelevant for the current time step. The forget gate takes the previous hidden state and current input as inputs and outputs a forget factor for each element in the cell state.

Output Gate: The output gate decides what the next hidden state should be and what should be output based on the current input and the updated cell state. It uses the sigmoid activation to decide which parts of the cell state should be included in the output and then applies the hyperbolic tangent function to squish the values to the desired output range.

How these components interact with each other and how the information, and signal flow through one LSTM unit is also displayed in Figure 3 (Data Base Camp, 2022). There, the information flows from left to right, with the inputs into the unit being the input x at timestamp t and the hidden and cell state from the previous time step. Then, the signal is passed through the

Figure 3
LSTM overview



forget gate, input gate, and output gate, before the hidden and the cell state of the current timestamp are calculated.

The key advantage of LSTMs lies in their ability to learn and adapt to long-range dependencies in sequences. The cell state allows them to store relevant context information over many time steps, and the gates enable them to control the flow of information into and out of the cell state, allowing them to remember or forget information as needed.

3.2. LSTM network for predicting leaks

The first goal of the SANDMAN solution is to predict whether there is a leak in a pipe. For this, a simple LSTM network is chosen, with 5 hidden LSTM layers and an input and output layer. The LSTM layers are in decreasing size, this enables the LSTM to create many hidden views of the data first. A large number of parameters are also useful to prevent overfitting as Belkin et al. (2019) have shown in their recent research.

To make sure all the parameters are utilized to learn the objective, a 50% dropout is applied after each LSTM layer. An overview of the networks structure can be found in Figure 4.

If the output of this network prediction is a leak, then the data is fed through another LSTM network which prediction the size and position of this leak.

3.3. LSTM network for predicting size and distance of a leak

This LSTM network for size and distance prediction is the second part of the SANDMAN solution. Here, a very similar but smaller neural network was chosen. This is because of a lack of data for this use case, and therefore faster convergence of the smaller network. Similar to the LSTM network from the leak prediction, a 50% dropout is applied after each LSTM layer. An overview of the networks structure can be found in Figure 5.

4. Experimental Setup

4.1. Testbed description and data collection

To train and test the LSTM network, an appropriate dataset was generated. To simulate different leakage conditions, an experimental measurement setup was put in place. It consists of

- Two pipes to simulate different vibration conditions
- Four taps distant from each other and with different size to simulate different leak conditions
- The measurement board containing the accelerometer sensor.

An overview of the test setup is depicted in Figure 6.

Figure 4
LSTM network for leak detection

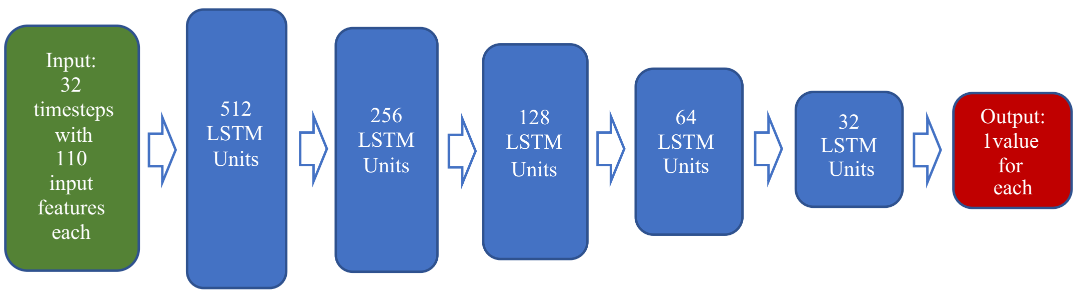
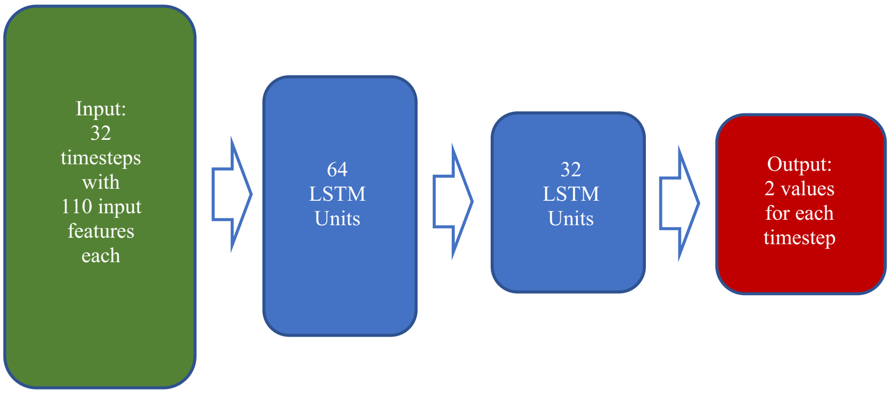


Figure 5
LSTM network for size and distance prediction



Using this test bed setup, data were collected in two test campaigns. The first run included several scenarios depicted in Table 1. Here, we can note that the first four scenarios are rather simple, like opening one specific tap slowly or abruptly. Scenarios 5–7 are more complex, because there are multiple leaks (tap openings) at once.

In the second test campaign, data were collected using two sensors and the water flow was inverted to the first test campaign and three scenarios were performed. First, all four taps were opened and closed one after the other. In the second and third tests baseline, vibration measurements were collected, where the vibration from the surroundings was measured and then the vibration from the pump.

Overall, these two test runs combined create a dataset with close to 47 million datapoints, each with one or two vibration values, and labels according to which taps were opened.

4.2. Data preprocessing

Before the vibration data can be used to train an LSTM network, some preprocessing needs to be done. First, the data have to be concatenated, so a single dataset from both test campaigns can be retrieved.

Then, the leaks are transferred into a distance and size values, based on the data shown in Figures 6 and 7. The distance will be measured in meters from the position of the sensor, in the direction of the flow. Because of this, sensor 1 of the second test campaign will have negative distances as the sensor is before the leaks. When multiple taps are opened, the bigger taps have the dominating value. Once the dataset is completed, the data can be preprocessed. For this, a more classical machine learning approach is chosen, where meaningful features are engineered first before a neural network is trained. In this case, a fixed time interval is defined, like half a second, and then takes the mean, squared mean, standard deviation, as well as frequency-based features of the time interval. This reduces the data significantly and helps the neural network to converge faster as meaningful features are already constructed.

After the features are created, the data will also be scaled, as this will make it easier for the neural networks to train on the data. Last, for both models, input sequences have to be created. The data are reshaped to be compatible with the architecture of LSTM networks. The input data to the LSTM network will be a three-dimensional tensor with the following form: *[data_samples, time_steps, features]*.

Once the preprocessing is done, the data will be used to train different models. For the first model, the goal is to predict whether there is a leak in the pipe. The architecture for this is described in Section 3.2. The second model will have the goal to predict the size and distance of the leak, with the architecture shown in Section 3.3.

5. Experiments Evaluation

5.1. Implementation details

The goals of the experiment were to be able to accurately predict whether there is a leak in a pipe and when there is a leak to accurately predict the size and the distance from the sensor. To address these goals, the above-mentioned approach will be used to train the LSTM networks.

Figure 6
Pipe's setup for data acquisition and experiments

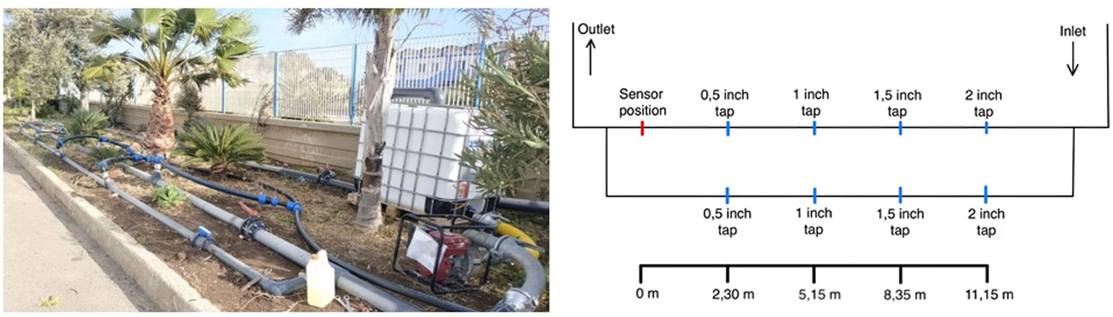
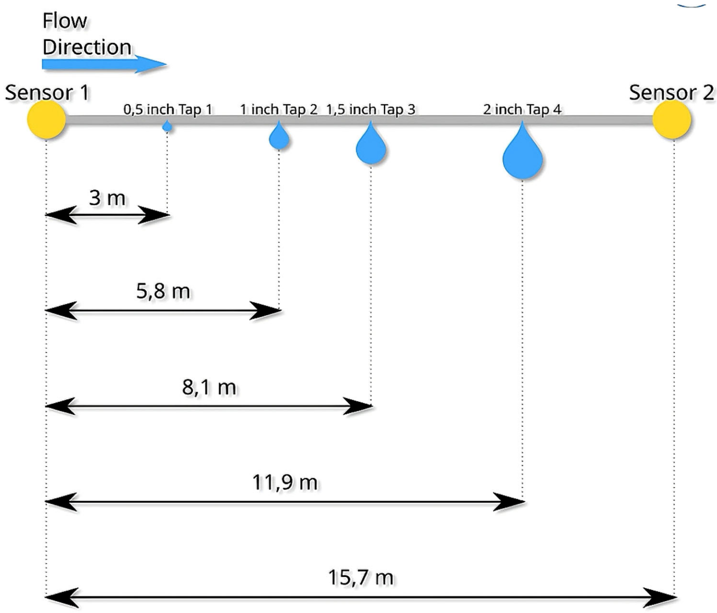


Figure 7
Setup and scenarios for the second test campaign

Sensor	Measurement Name	Procedure	Tag
1	Leakage	1 min	AllTapsClosed
		2 min leakage 4	Tap4Open
		1 min	AllTapsClosed
		2 min leakage 3	Tap3Open
		1 min	AllTapsClosed
		2 min leakage 2	Tap2Open
		1 min	AllTapsClosed
		2 min leakage 1	Tap1Open
		1 min	AllTapsClosed
1	Scenario1	3 min	PumpOff
1	Scenario2	3 min	PumpOn
2	Leakage	1 min	AllTapsClosed
		2 min leakage 4	Tap4Open
		1 min	AllTapsClosed
		2 min leakage 3	Tap3Open
		1 min	AllTapsClosed
		2 min leakage 2	Tap2Open
		1 min	AllTapsClosed
		2 min leakage 1	Tap1Open
		1 min	AllTapsClosed
2	Scenario1	3 min	PumpOff
2	Scenario2	3 min	PumpOn



As mentioned, this approach relies heavily on preprocessing the data, before training a neural network with this preprocessed data. Before any features could be engineered, the dataset had to be constructed. For this, a dataset from the first test campaign (setup shown in Table 1 and Figure 6) was merged with the second test campaign (shown in Figure 7). Because the second test campaign has two sensors that recorded the data, the data from each sensor were appended to the dataset. So, a dataset with always only one measurement is retrieved. On this base dataset, the feature engineering is done. To achieve this, an interval of 500 ms was chosen and then summarized by taking certain properties of that interval. The features created were chosen in such a way to have a high correlation with leakage detection and position. The features chosen for the final implementation were the squared mean, the standard deviation, and the minimum and maximum values. Also, frequency dependent features were chosen. For this, fast-Fourier-transform was applied and then of each 100 Hz interval the standard deviation, minimum, and maximum values were taken. As the sampling rate of the sensor was 6700 Hz, this resulted in 109 features. This is a big reduction from the 3500 sensor values recorded in this 500 ms timeframe.

After the features are created, the data were scaled between 0 and 1. This leads to a dataset of 16365 rows and 109 columns. The dataset was then split into train and test dataset. The test dataset contains runs 5, 6, and 7 from Table 1 and the beginning of each leakage in Figure 7. The training dataset includes all the rest. This results into 10756 rows used for training. It is important to mention that due to the relatively short time, the taps were opened in the second test campaign as shown in Figure 4; there are much less data available for this test campaign. In numbers, there are only 1630 values belonging to each sensor in the second test campaign, while 1150 of these values are used for training. Compared to the 7500 values that are used for training from the first test campaign, it becomes clear that the model will have much more difficulties to learn the behavior of the second test campaign. Because of this and the big difference between the two test campaigns, the results will be measured separately for each test campaign.

To start creating a model, an unsupervised anomaly detection approach was tested. For this, a LSTM autoencoder was trained on 6282 rows of the dataset because all the rows where the tap was opened were removed, so the autoencoder only learns what normal (not leaky) data look like. The approach for the autoencoder was not successful. The reason for this is the fact that the autoencoder is able to reconstruct the data where the tap is opened very well and does not have a bigger reconstruction error.

Because of this issue, a supervised approach was chosen, with one predicted feature, which says whether there is a leak in the pipe with the mentioned 10756 rows used as a training dataset, while the rest is used as a test dataset, on which the results will be measured.

According to some recent research (Belkin et al., 2019), large neural networks are less prone to overfitting and generalize better. Because of this, a large neural network with 5 LSTM layers, 1 output layer, and overall, 2,320,289 parameters were trained for about 1000 epochs.

The models were trained on the platform Kaggle[3], which is a platform to write python jupyter notebooks online. The advantage is the models can be trained on their dedicated hardware. To train the models, CPUs, GPUs, and TPUs are available. To get the fastest training results, it is best to use the TPUs, but as they are only limited TPUs available and sometimes long-waiting times, most of the models, which used the preprocessed features, were trained on GPUs. The GPUs were only slightly slower than the TPUs. In numbers, the training on the CPU takes about 1 second per step, while the training on the GPU takes around 60 milliseconds and on the TPU takes around 45 milliseconds per step.

Besides detecting whether there is a leak in the pipes, another objective was to learn to predict the size and distance of a leak. For this, the data were filtered and all the datapoints without a leak were removed. Despite this, all the steps from before were repeated with the only difference that the predicted features, in this case, contain two variables, one for the size and one for the distance. The distance can be negative as well, so the leak is predicted in a distance with the direction of flow.

[3]https://www.kaggle.com/

Table 1
Recorded data scenarios for the first test campaign

Scenario	Conditions							Starting Time
0	All Taps Closed (ATC)							12:03
	9 min 57 s							
1	ATC	Tap1AB	ATC					13:00
	3 min 2 s	3 min 5 s	2 min 18 s					
1a	ATC	Tap1SLOpening	Tap1Open	ATC				13:18
	2 min	31s	1 min 33 s	2 min 8 s				
2	ATC	Tap2AB	ATC					13:25
	3 min 16 s	2 min 46 s	1 min 46 s					
2a	ATC	Tap2SLOpening	Tap2Open	ATC				15:17
	3 min 14 s	27 s	1 min 45 s	2 min 13 s				
3	ATC	Tap3AB	ATC					15:25
	3 min	2 min 4 s	2 min 10 s					
3a	ATC	Tap3AB	ATC					15:33
	3 min 11 s	2 min 6 s	2 min 11 s					
4	ATC	Tap4AB	ATC					15:41
	3 min 11 s	2 min 20 s	2 min 8 s					
5	ATC	Tap2AB	Tap2Open + Tap1AB	Tap1Open	ATC			15:49
	3 min 4 s	2 min 9 s	2 min 11 s	2 min 13 s	2 min 5 s			
6	ATC	Tap4AB	Tap4Open + Tap1SLOpening	Tap1Open + Tap4Open	Tap1Open	ATC		16:04
	3 min 2 s	2 min 15 s	15 s	1 min 56 s	2 min 59 s	2 min 12 s		
7	ATC	Tap4AB	Tap4Open + Tap2AB	Tap2Open + Tap4Open + Tap1AB	Tap1Open + Tap2Open	Tap1Open	ATC	16:22
	3 min 30 s	1 min 16 s	1 min 10 s	1 min 1 s	1 min 3 s	1 min 12 s	1 min 2 s	

After the values are fed through the network, a prediction is received for each 0.5 s, but because there is quite a lot of variance and some outliers, only the median value of each 10 s will be taken, and all the other values will not be used to measure the results.

5.2. Experiments results

With the supervised approach, all the leaks can be detected although sometimes the model also detects a leak when there is none. We believe the reason for this behavior is probably a lack of data. To go more into detail, a look at the test scenarios and the training data needs to be taken. As mentioned, there are three scenarios 5, 6, and 7 from Table 1 which are completely used as test data. These scenarios are from the first test campaign, with a lot of training data available as scenarios 1–4 are used for training the network. The results of the algorithm for scenario 5 can be seen in Figure 8. It shows the predicted (blue) and measured (red) values. One can see that the network predicts the leak, quite accurately, but there are three positions which need to be discussed, as shown by the dotted squares. In the first square, it takes the network 24 s to realize a tap was opened for the second dotted box, the network shortly thinks that the tap was closed again. This is not true, but from reviewing Table 1 again, it seems around this time an extra tap was opened, so maybe this caused a change in the data, thus leading the network to the wrong prediction. The same thing is likely what happened in the third dotted square. These spikes do not represent a major issue, as long as the network is generally correct, and only few of those appear because it is still pretty clear from the blue line to see at which time a tap was opened.

For the sixth scenario, it seems similar problems appear, and they look more severe than in scenario 5. As shown in Figure 9, it is still possible to see that there is quite different behavior when the taps are opened, especially since there are no false positives. Still during the time the taps were indeed opened, the network predicts this more than half of the times, and just reviewing the blue line an operator would be quite certain there is a leak around 06:04. This is a 68 s delay from when the leak actually occurred.

Similar results are produced by scenario 7, shown in Figure 10. There, the network still detects the leakage but only after a long time of 110 s, the score seems to be conclusive.

The results for these three test runs are also shown in the second column of Table 2. In the other two columns, the results for the size and distance prediction are shown. For this, a much smaller LSTM network with only 58.000 parameters was trained. The network size was reduced because the training set for this use case is even smaller, as only data with an opened tap were used. This resulted in a training set of only 3235 rows. Nonetheless, the results show that the approximation is quite accurate, with the median absolute error of the scenarios of the first test campaign smaller than 0.04 inch for the size, and the distance prediction with an accuracy of 0.5 m. The median absolute error was chosen in this case because there are sometimes outliers in the prediction as it can be seen in an exemplary plot for scenario 7 shown in Figure 11.

Besides these three scenarios from the first test campaign, the leakage scenario from the second test campaign shown in Figure 7 was also used to test the algorithm. But as mentioned earlier, since the scenario has such little data, the results related to this scenario are unsatisfactory as expected and the results will be displayed separately. As this was the only scenario with a leak, in this test campaign, this scenario was split up into training and test sets, with the time around the first occurrence of a leak used only for testing. Because the data were recorded with two sensors, a

Figure 8
Results for scenario 5

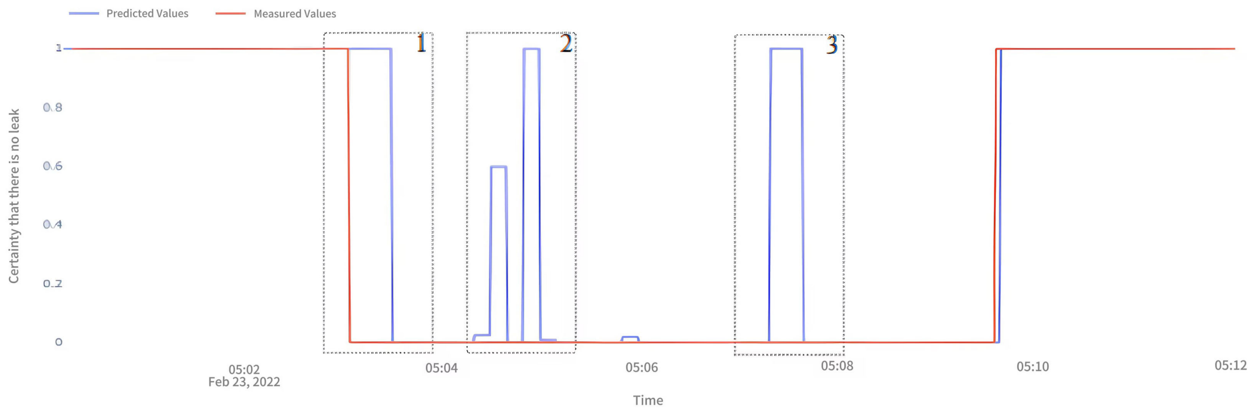


Figure 9
Results for scenario 6

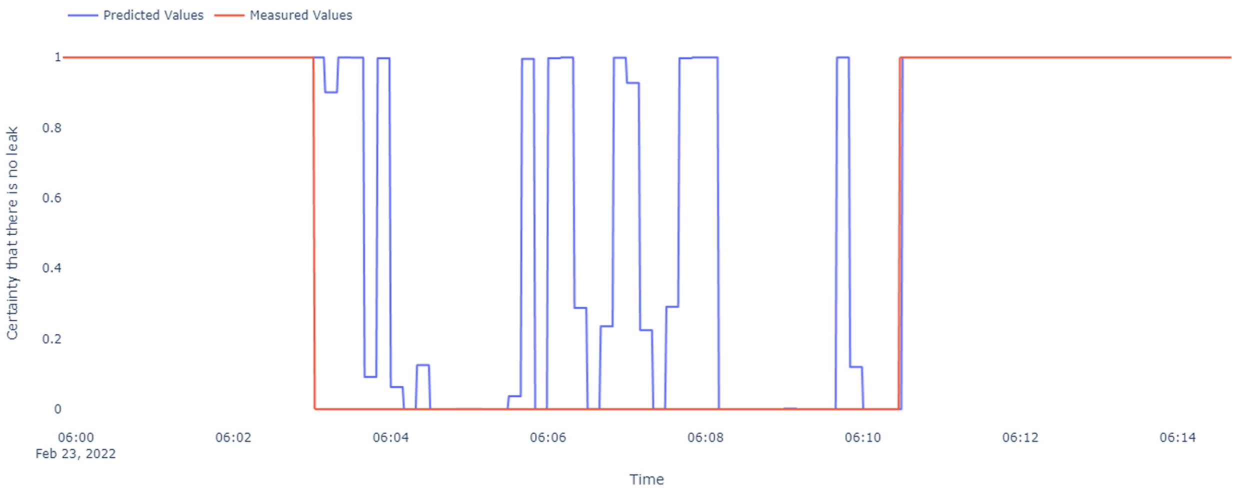
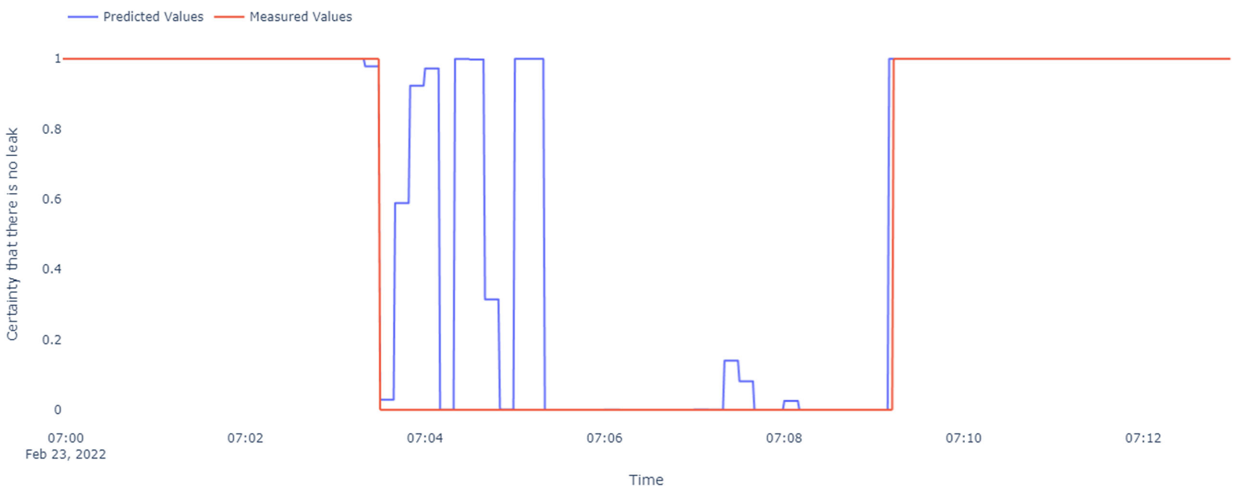


Figure 10
Results from scenario 7



picture for each sensor is shown. Figure 12 shows the leak detection results for sensor 1. With these results, an operator could potentially predict a leak in the first and the third of these graphs. But the second and especially the fourth leak are problematic. The main problem in the fourth graph is that in the beginning the network predicts a leak, although there is no leak. This has not happened in any of the cases before and makes it much harder to decide if there is a leak or not, without the measured values.

Table 2
Results from first test campaign

Scenario	Time to recognize leak in s	Leak size prediction median of absolute error in inches	Leak distance prediction median of absolute error in meter
5	24	0.03	0.15
6	68	0.03	0.43
7	110	0.002	0.39

For the other sensor in this scenario, the data look quite similar, with accurate leak detection in the first and third plots of Figure 13 and inaccurate detection for the second and fourth plots because of false positives.

The results for the leak detection are again summarized in Table 3.

As shown in Table 3, the results for the size and distance prediction of the second test campaign are much less encouraging. This is once again probably due to the lack of training data because each tap was only opened for 2 min leading to 180 training samples per tap as 60 samples were used for testing. With a larger dataset, better results could probably also be achieved there. Another notable point about this test case is especially the fact that for sensor 1, the results get better with the leak being closer to the sensor because tap 4 is the tap that is the furthest from sensor 1, while tap 1 is the closest. For sensor 2, this is not the case.

To sum this section up, leakage detection and size and distance prediction is quite accurate, for the first test campaign with much training data, but on the other hand, the same models struggle with these tasks when there is only little data available as in the second test campaign. Once it was detected, there is a leak in the pipe; this leak can then be predicted according to distance and size of the leak. Both these functionalities are highly dependent on the training data and a sufficient training set needs to be created. This was devised by the relatively bad results for the scenario "leakage" in Table 3 compared to the other scenarios in Table 2.

Table 3
Results from second test campaign

Scenario	Time to recognize leak in s	Leak size prediction median of absolute error in inches	Leak distance prediction median of absolute error in meter
Leakage 4 sensor 1	13	0.75	7.09
Leakage 3 sensor 1	Not possible	0.34	3.07
Leakage 2 sensor 1	2	0.38	2.14
Leakage 1 sensor 1	Not possible	0.14	0.36
Leakage 4 sensor 2	33	0.32	4.25
Leakage 3 sensor 2	Not possible	0.21	1.24
Leakage 2 sensor 2	2	0.16	2.34
Leakage 1 sensor 2	Not possible	1.33	4.15

Figure 11
Leak size and distance prediction results for scenario 7

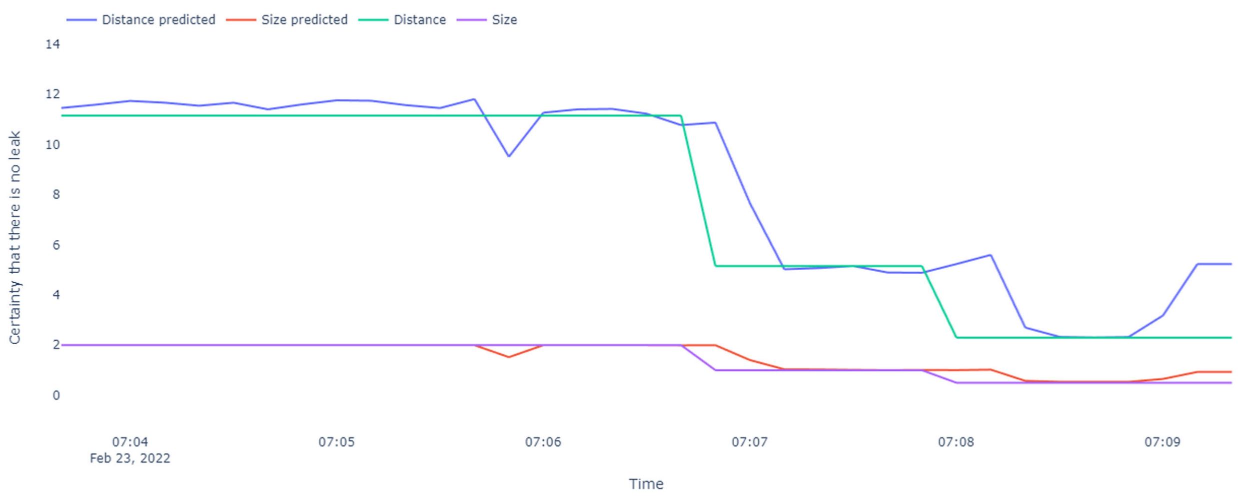


Figure 12
Leakage detection for sensor 1 of leakage scenario of Figure 7

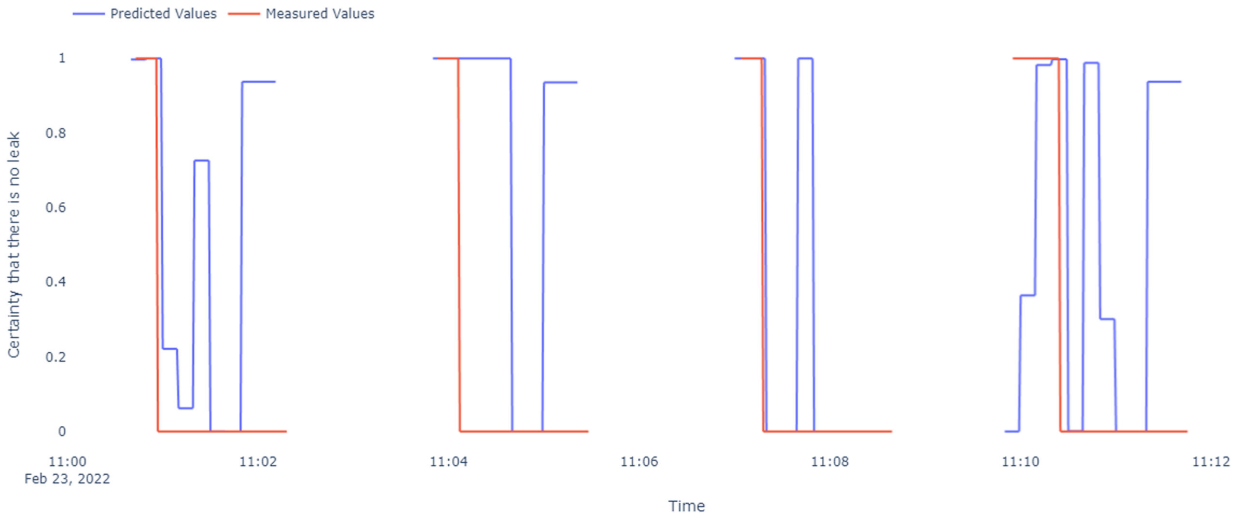
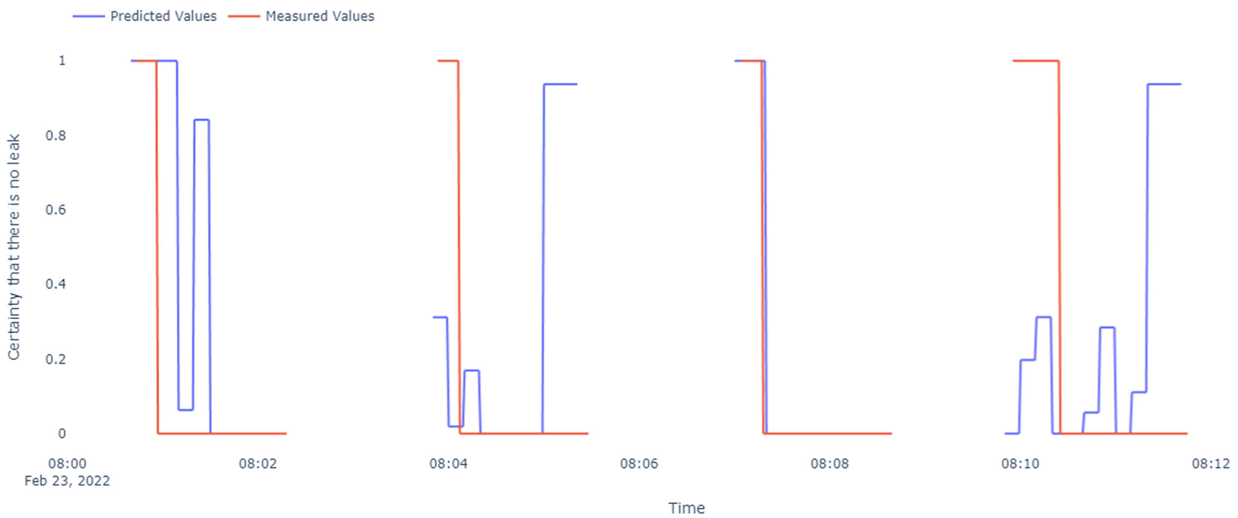


Figure 13
Leakage detection for sensor 2 of leakage scenario of Figure 7



6. Conclusion

In this paper, we have proposed a solution for dealing with water waste due to leakage in pipes as it is becoming a real menace for our daily life. To be more concrete, we have discussed a predictive maintenance solution for smart pipes. We have argued that incorporating appropriate sensors (such as vibration sensors) into water pipes networks can lead to predicting potential water leakage early enough and in an accurate way. The pipes conditions are continuously monitored, and the collected sensors data are analyzed using deep learning techniques. Our solution is validated through an experimental layout put in place in the EKSO S.R.L premises in Italy. The testbed is composed of two pipes where vibration sensors are integrated and a number of taps that simulate the leaks when they are open. Our experiments, in particular, the developed LSTM models, have shown that the detection of the leaks, their sizes, and their distances from the sensors are quite accurate when enough training data are used. But before this technology can make its way into industry, further testing is needed with the creation of bigger and more diverse dataset with more diverse leak sizes and positions, and with test pipes underground.

It is expected that the developed solution significantly contributes to the preservation of water resources, mitigates the effects of water scarcity, and minimizes the carbon footprint associated with water treatment and transportation.

Funding Support

This work has been carried out within the SANDMAN project. The latter is an experimental solution accepted and undertaken within the second open call[4] of the H2020 AIREGIO project[5] (Grant Agreement Number 952003). The SANDMAN project explores AI-based predictive maintenance for smart pipes. The AIREGIO project aims at filling major gaps currently preventing AI-driven Digital Innovation Hubs (DIH) from implementing fully effective digital transformation pathways for their Manufacturing SMEs.

[4]https://www.airegio-project.eu/open-call-2
[5]https://www.airegio-project.eu/

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data available on request from the corresponding author upon reasonable request.

References

Alawadhi, A., & Tartakovsky, D. M. (2020). Bayesian update and method of distributions: Application to leak detection in transmission mains. *Water Resources Research*, *56*(2), e2019WR025879. https://doi.org/10.1029/2019WR025879

Almobarek, M., Mendibil, K., & Alrashdan, A. (2023). Predictive maintenance 4.0 for chilled water system at commercial buildings: A methodological framework. *Buildings*, *13*(2), 497. https://doi.org/10.3390/buildings13020497

American Society of Civil Engineers. (2021). *Report card for AMERICA's infrastructure*. Retrieved from: https://infrastructurereportcard.org/cat-item/drinking-water-infrastructure/

Belkin, M., Hsu, D., Ma, S., & Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–Variance trade-off. *Proceedings of the National Academy of Sciences*, *116*(32), 15849–15854.https://doi.org/10.1073/pnas.1903070116

Cardoso, A., Oliveira, J., Neto, D., Fernandes, M., Petrella, L., Henriques, J., . . . , & Rebahi, Y. (2023). Technical components integration using APIs for predictive maintenance in the context of Industry 4.0 digital transformation. In *International Conference on Remote Engineering and Virtual Instrumentation*, 951–960. https://doi.org/10.1007/978-3-031-42467-0_89

Carvalho, T. P., Soares, F. A., Vita, R., Francisco, R. D. P., Basto, J. P., & Alcalá, S. G. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, *137*, 106024. https://doi.org/10.1016/j.cie.2019.106024

Cinelli, A. (2023). *Italy's water crisis exacerbated by leaky pipes, data shows*. Retrieved from: https://www.reuters.com/business/environment/italys-water-crisis-exacerbated-by-leaky-pipes-data-shows-2023-03-21/#:~:text=ROME%2C%20March%2021%20(Reuters),bureau%20ISTAT%20said%20on%20Tuesday

Data Base Camp. (2022) *Long short-term memory networks (LSTM)-Simply explained!* Retrieved from: https://databasecamp.de/en/ml/lstms

Gebelhoff, R. (2023). *A mostly hidden problem wastes appalling amounts of water*. Retrieved from: https://www.washingtonpost.com/opinions/2023/03/27/water-utilities-infrastructure-leaks-conservation/

Gorenstein, A., & Kalech, M. (2022). Predictive maintenance for critical infrastructure. *Expert Systems with Applications*, *210*, 118413.https://doi.org/10.1016/j.eswa.2022.118413

Hilliger, B., Rebahi, Y., Lepadatu, B. A., & Cardoso, A. (2023). Deep learning based anomaly detection for predictive maintenance. In *3rd IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0/5.0*, 56–62.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Kamat, P., & Sugandhi, R. (2020). Anomaly detection for predictive maintenance in industry 4.0 – A survey. *E3S Web of Conferences*, *170*, 02007. https://doi.org/10.1051/e3sconf/202017002007

Karimli, F. (2023). *The last drop of water: The European water crisis*. Retrieved from: https://europeanstudiesreview.com/2023/04/25/the-last-drop-of-water-the-european-water-crisis/

Lowin, M., & Mihale-Wilson, C. (2021). Towards predictive maintenance as a service in the smart housing industry. *Lecture Notes in Informatics*, 1093–1106. https://doi.org/10.18420/informatik2021-089

Lughofer, E., & Sayed-Mouchaweh, M. (2019). *Predictive maintenance in dynamic systems: Advanced methods, decision support tools and real-world applications*. USA: Springer.

Mobley, R. K. (2002). *An introduction to predictive maintenance*. UK: Butterworth-Heinemann.

Rebahi, Y., Hilliger, B., Lowin, P., & Cardoso, A. (2023). AI based predictive maintenance as a key enabler for circular economy: The KYKLOS 4.0 approach. In *19th International Conference on Distributed Computing in Smart Systems and the Internet of Things*, 367–372. https://doi.org/10.1109/DCOSS-IoT58021.2023.00066

Sadeghioon, A. M., Metje, N., Chapman, D. N., & Anthony, C. J. (2014). SmartPipes: Smart wireless sensor networks for leak detection in water pipelines. *Journal of Sensor and Actuator Networks*, *3*(1), 64–78. https://doi.org/10.3390/jsan3010064

Sharma, J., Mittal, M. L., & Soni, G. (2022). Condition-based maintenance using machine learning and role of interpretability: A review. *International Journal of System Assurance Engineering and Management*. https://doi.org/10.1007/s13198-022-01843-7

Sohaib, M., Mushtaq, S., & Uddin, J. (2021). Deep learning for data-driven predictive maintenance. In M. A. R. Ahad & A. Inoue (Eds.), *Vision, sensing and analytics: Integrative approaches* (pp. 71–95). Springer. https://doi.org/10.1007/978-3-030-75490-7

Tercan, H., & Meisen, T. (2022). Machine learning and deep learning based predictive quality in manufacturing: A systematic review. *Journal of Intelligent Manufacturing*, *33*(7), 1879–1905. https://doi.org/10.1007/s10845-022-01963-8

Virk, M. U. R. A., Mysorewala, M. F., Cheded, L., & Ali, I. M. (2020). Leak detection using flow-induced vibrations in pressurized wall-mounted water pipelines. *IEEE Access*, *8*, 188673–188687. https://doi.org/10.1109/ACCESS.2020.3032319

Zhu, T., Ran, Y., Zhou, X., & Wen, Y. (2019). A survey of predictive maintenance: Systems, purposes and approaches. *arXiv Preprint:1912.07383*.

How to Cite: Rebahi, Y., Hilliger, B., Lowin, P., Zheng, B., Bormida, G. D., & Ladjeri, K. (2024). Toward AI-Based Condition Monitoring and Predictive Maintenance for Water Smart Pipes: The SANDMAN Approach. *Artificial Intelligence and Applications*, *2*(2), 86–96. https://doi.org/10.47852/bonviewAIA32021513