Digit. Signal Process. 168 (2026) 105603

Contents lists available at ScienceDirect

Digital Signal Processing

journal homepage: www.elsevier.com/locate/dsp

Moving forward in water distribution network leak identification through an innovative features engineering step

Elvio Damonti *, Giancarlo Bernasconi

DEIB (Dipartimento di Elettronica, Informazione e Bioingegneria), Politecnico di Milano, Milan, Italy

ARTICLE INFO

Keywords:
Leak identification in Water Distribution Networks (WDN)
Water pressure data features engineering
Machine learning methods for water leak detection and localization
Convolutional neural network (CNN)
Overcomplete Autoencoder application
CNN Overcomplete Autoencoder inference errors analysis

ABSTRACT

In Italy an average of 40% of the water in transit in Water Distribution Networks (WDNs) is lost, which represents a significant economic loss, and, because of the corresponding energy waste, also a considerable environmental damage. Over the years various methods have been developed for detecting and locating leaks in WDNs, and, in particular, several algorithms have been implemented that use Convolutional Neural Networks (CNNs). They are all based on a training phase run on a representative subset of the WDN data, and the main differences between the various implementations are in the data pre-processing and in the CNN configuration. This paper proposes a new fully Data-Driven approach, where a preliminar features engineering step, performed by a visual analysis of specific data patterns, both in the time domain and in the Fourier domain, allowed us to conceive and identify two paramount features engineering steps: a new effective data pre-processing algorythm and a new configuration of CNN that uses an Overcomplete Autoencoder (Overcomplete AE) topology with residual blocks. These two steps, described in detail in this paper, allowed us to better highlight and identify the anomalies caused by leaks in WDN pressures time series and they permitted, in association with a new original automatic analysis of the reconstruction error made by the Autoencoder, to achieve results that are on the top of the current state of art. Specifically, the whole innovative method is presented in detail exploiting publicly available data, so to be easily reproducible, and, more specifically, for this purpose, the benchmark was run on a synthetic 'LeakDB' dataset of 500 scenarios and the outcomes were then validated through different data obtained from a second and more complex synthetic 'LeakDB' dataset containing 1000 scenarios. Both these datasets are related to a District Metering Area (DMA) of the Hanoi WDN and both are publicly available.

1. Introduction

In recent years, global warming phenomena have exacerbated the issues related to the availability of drinking water. Its increased demand is not fully met in many areas, also caused by the aging and deterioration of Water Distribution Networks (WDNs) and to the consequent leaks. All these factors have led to intensified research aimed at reducing leaks in WDNs, which in Italy average 40 %, but similar issues are also present to varying degrees in other countries [1]. These leaks represent a significant economic loss and, caused by the corresponding energy waste, also a considerable environmental impact [2–4].

Over the years, various methods have been developed for detecting and locating leaks in WDNs using, in general, pressure and flow rate as measured input variables [5–15]. However, flow sensors are more expensive and require more costly installation and maintenance procedures, thus, most leak detection methods in WDNs assume only the measurement of pressure, with ideally a pressure sensor at each WDN node collecting data as frequently as possible. One objective thus becomes also to find out methods to reduce the number of pressure sensors without loosing the ability to detect and localize the leaks. To aim of detecting and locating leaks the literature currently presents methods based on transient analysis [5,15] and methods based on steady-state analysis [6,7,12,16]. For steady-state analysis there are model-based methods [6,7] and data-driven methods [8,16], where data-driven methods, including the one we are proposing, have the advantage of to not require any knowledge of the morphological characteristics (length, diameter, configuration, roughness, elevation, curvature, junctions of the pipes or of the other components of the WDN) [10]. All these methods exploit the temporal sequences of data (mainly pressure data) collected in the WDN [11] and, only if leak localization is required in addition to detection, they need the knowledge of the WDN's topology. In these methods, the pressure variations caused by leaks are

* Corresponding author.
E-mail addresses: elvio.damonti@polimi.it (E. Damonti), giancarlo.bernasconi@polimi.it (G. Bernasconi).

https://doi.org/10.1016/j.dsp.2025.105603

Available online 9 September 2025

always seen as anomalies in the temporal sequences of pressure data and thus the problem of detecting leaks is reduced to problem of identifying anomalies within these sequences. Among data-driven methods several use neural networks and, specifically, algorithms based on Convolutional Neural Networks (CNNs) [3,16,17] and/or on Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) [18], or Gated Recurrent Networks (GRU) [19]. T. K. Chan, Cheng Siong Chin, and Xionghu Zhong [20] and Yipeng Wu and Shuming Liu [21] wrote a valid reviews of current technologies and methodologies for Water Distributed Network leakage detection, some extracts of which are presented in the 'Conclusions' paragraph of our paper.

We present here a new configuration for a CNN that exploits an Overcomplete Autoencoder (Overcomplete AE) topology, with many layers and several 'residual blocks', suitable to detect also minimum variations of patterns in pressures time series and to better generalize the provided neural model against the seasonal variations of WDN water demands in different scenarios. Unfortunately, until now, we have not been able to obtain real data from a real WDN and therefore our innovative method has been tested and validated only on some synthetic data publicly available and used by many researchers as benchmarks for their leak detection procedures [22,23]. Using the same 'standard' datasets makes it easy to compare the performance and peculiar characteristics of various leak detection methods. Several similar pressure datasets were generated and organized, consisting each one of a variable number of scenarios, by researchers at the 'KIOS Research and Innovation Center of Excellence (KIOS CoE)' [22] and, specifically, in this paper we use, among these ones, the specific dataset of 500 scenarios (380 with leaks and 120 without leaks) [22,23] and, for validation purpose, 1000 scenarios from a different and more complex specific dataset of 1000 scenarios (762 with leaks and 238 without leaks) [22,23]. The synthetic pressure data included in these datasets were generated with reference to a part, i.e. a District Metering Area (DMA), of the WDN of the city of Hanoi, which has 32 junctions and an elevated reservoir that supplies the WDN. For each day and for each dataset, 48 pressure samples are available for each junction of the WDN.These samples are spaced by 30 min and each samples scenario can refer either to a WDN state characterized by leaks or to a WDN state without leaks.

Regarding the neural networks, we decided to implement an innovative CNNs in an 'Overcomplete Autoencoder' configuration; where an Autoencoder consists of a part named Encoder, followed by a part named Decoder [24,25]. The Encoder part generally reduces the dimensionality of the input data (undercomplete Autoencoder), compressing them to extract the most significant features, while instead, generally, the Decoder part tries to reconstruct the input data from the compressed output of the Encoder part. Through these operations an Autoencoder can reconstruct the input data with limited errors when the patterns of input data are similar to those used in its training, but it generates much larger reconstruction errors when the input data have instead patterns sufficiently different from those used during its training trial [25,26]. Advantageously, for the aim of leaks detection, this reconstruction error caused by anomalies in the input data, is amplified in an 'Overcomplete Autoencoder', where the Encoder part increases the dimensionality of the input instead of reducing it and where the Decoder part reduces it back to the original value.

Most steady-state leak detection methods use pressure (and/or flow rate) data collected during times of the day with minimal stochastic variance: typically, during Minimum Night Flow (MNF) [12,22]. Among the most significant works of this kind, Lourierio et al. in [13], proposed a four step approach to reduce the amount of required data, while Soldevila et al. in [12], used flow rate measurements at the inlet of a single zone (DMA) of the WDN [14]. These authors then solved the problem of localizing the detected leak using clustering techniques. On the contrary, our method uses the data collected throughout the entire day allowing more comprehensive statistical data collections and the exploitation of all data recorded by the sensors.

Our method involves the following main steps:

- pre-processing of pressure data time series;
- using of pressure data from two leak-free scenarios to train a 1D CNN in 'Overcomplete AutoEncoder' configuration;
- processing of pressure data from other scenarios from each dataset, both with and without leaks, using the above trained 1D CNN;
- comparing the original input data fed into this trained 1D CNN with the output of the same neural network to detect leak scenarios and localize the WDN junctions where leaks occur.

The following sections of the paper explain in detail the characteristics of the problem we aim to solve, the innovative method we propose and the results of our simulations and of our work.

2. Leaks detection

2.1. The problem to solve

In a WDN (Water Distribution Network), assuming that water consumption at the WDN junctions and the state of all other elements of the WDN remain constant, the presence of a leak causes a reduction in pressure near the leak. Unfortunately, even in the absence of leaks, the pressure at various points within the WDN continuously fluctuates randomly, because of unknown and random changes in the WDN's morphology (such as variations in the size of the WDN, pipe roughness, valve conditions, pump conditions, and the elevation of water in reservoirs) and because of unknown and random changes in water withdrawals (water demands) from various WDN nodes. Moreover these variations in withdrawals can occur instantaneously, or with hourly, or with daily, or with seasonal periodicity, and they are always characterized by stochastic variance relative to their nominal values.

The goal of our work, in spite of these pressure fluctuations not caused by leaks, is to detect the existence of leaks in the Hanoi WDN's DMA (hereafter referred to simply as WDN) and to locate the specific junctions, within the WDN, where leaks are present.

2.2. Our method

To detect and locate leaks in the WDN (see Fig. 2), a specific 1D CNN neural network model in an 'Overcomplete Autoencoder' configuration was generated, trained and used, for each junction of the WDN. Each junction of the WDN has been assumed to be equipped with a pressure sensor that, synchronously with the sensors of the other junctions, samples the pressure signal. For each junction of the WDN pressure data, over a sufficiently long period (in our case this period matches the scenario duration, i.e. one year), have been pre-processed and then organized into a row of a matrix $\mathbf{P}$. At each sampling instant $\mathbf{n}$, similar to what other researchers have done [16], $\mathbf{K}$ samples $\mathbf{p_k}$ are collected from the pre-processed pressure data matrix $\mathbf{P}$ and inserted in a column of a matrix $\mathbf{I^n}$; one for each one of $\mathbf{K}$ junctions of the WDN (where $\mathbf{p_k(n)}$ is the pre-processed pressure sample measured at instant $\mathbf{n}$ at junction $\mathbf{k}$).

Thus, these $\mathbf{K}$ samples $\mathbf{p_k}$ form the column $\mathbf{n}$ of a matrix $\mathbf{I^n}$ of pre-processed pressures and $\mathbf{I^n}$ is the matrix to be fed to CNNs (see *Eq. (1)*) whose content depends on the considered instant $\mathbf{n}$. The number of columns in the pre-processed pressure matrix $\mathbf{I^n}$, as proposed by another research group [16], has been fixed to the number $\mathbf{L}$ of samples collected for each junction over one week. This setup ensures robustness against seasonal pressure variations in the WDN [16], while still allowing for a temporal sequence samples of pressure data long enough to achieve high accuracy.

$$\mathbf{In} = \begin{bmatrix} \mathrm{p1(n)} & \mathrm{p1}(n-1) & \dots & \mathrm{p1}(n-L+1) \\ \mathrm{p2(n)} & \mathrm{p2}(n-1) & \dots & \mathrm{p2}(n-L+1) \\ \dots & \dots & \dots & \dots \\ \mathrm{pk(n)} & \mathrm{pk}(n-1) & \dots & \mathrm{pk}(n-L+1) \end{bmatrix} \tag{1}$$

At the instant $\mathbf{n}$, when a new column of samples enters the matrix $\mathbf{I^n}$, the column corresponding to the oldest samples is discarded. In such a

way the matrix $\mathbf{I^n}$ has fixed dimensions **K x L**, with each row corresponding to a junction in the WDN and with each column corresponding to a sampling instant. Each row of $\mathbf{I^n}$ is then processed as a single temporal sample by the corresponding 1D CNN neural network ('Overcomplete Autoencoder') of the junction it refers to and these individual temporal samples overlap for **L-1** pressure data points. In other words, the temporal samples processed by each neural network are obtained from a sliding window of length **L** on the pressure data in matrix **P**.

The 1D CNN neural networks bank (see Fig. 1), for each junction, outputs **L** pressure data values at each instant **n**, allowing the construction of a 2D output matrix $\hat{\mathbf{I}}^{\mathbf{n}}$ having the same dimensions **K x L** as the input matrix $\mathbf{I^n}$.

Essentially, the pre-processed pressure data matrix $\mathbf{I_n}$ is used as input for a bank of **K** 1D CNN 'Overcomplete Autoencoder' neural networks that operate in parallel, processing all the **K** rows of the matrix $\mathbf{I^n}$ at each instant **n** and generating a corresponding matrix $\hat{\mathbf{I}}^{\mathbf{n}}$ of predicted pressures data having dimensions **K x L**. To detect and locate potential leaks in the WDN, each row of the pre-processed actual pressures matrix $\mathbf{I^n}$ is then compared with the corresponding row of the output predicted (reconstructed) pressures matrix $\hat{\mathbf{I}}^{\mathbf{n}}$.

2.2.1. *The DMA (District metering area) of the WDN (Water distribution network) of hanoi*

The DMA of the WDN of Hanoi used in this paper (see Fig. 2) consists of 33 nodes (1 reservoir and 32 junctions) and has been modeled using the Epanet software [27,28], therefore, the number **K** of rows in the matrix of pre-processed pressure data $\mathbf{I^n}$ is equal to 32 and variable withdrawals are made from each junction of the WDN, both periodically and instantaneously.

2.2.2. *The pressures datasets*

The pressure datasets we used were generated [22,23] using publicly available libraries of standard sofware 'Epanet' [27,28] by introducing random values for the aforementioned variations in withdrawals for each scenario and for each junction in the Epanet model, publicly available, of the above DMA of Hanoi WDN [22,23]. For a given model of the WDN and for a known set of instantaneous withdrawals (water demans) Epanet calculates, under steady-state conditions, both the pressures at all nodes of the WDN and the flows in all its pipes. The first pressure dataset we used [22,23], generated in such a way, consists of **500** scenarios (**380** with leaks and **120** without leaks), each corresponding to the time series of pressures in the WDN 'measured' over one year and sampled at a frequency of 30 min: therefore, for each scenario and for each junction of the WDN, we have **48** pressure samples per day, **336** samples per week and **17,520** samples for year. The second different and **more complex** dataset we used [22,23] for validation purpose is generated in a similar way, but with different and smaller leaks and different WDN parameters values, and consists of **1000** scenarios (**762** with leaks and **238** without leaks). The number **L** of columns in the above pre-processed pressure data matrix $\mathbf{I^n}$ is thus equal to **336**. Each of these samples is, as previously emphasized, affected by huge

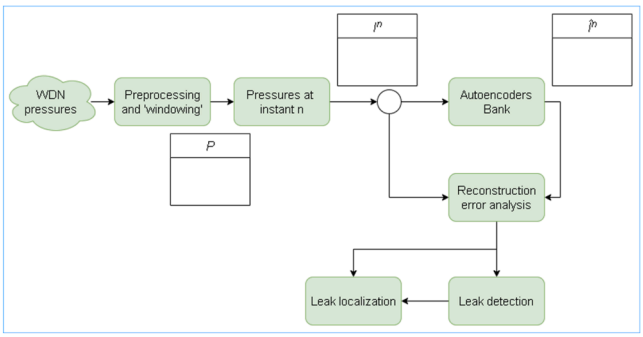

Fig. 1. block diagram of the method for detecting and locating leaks.

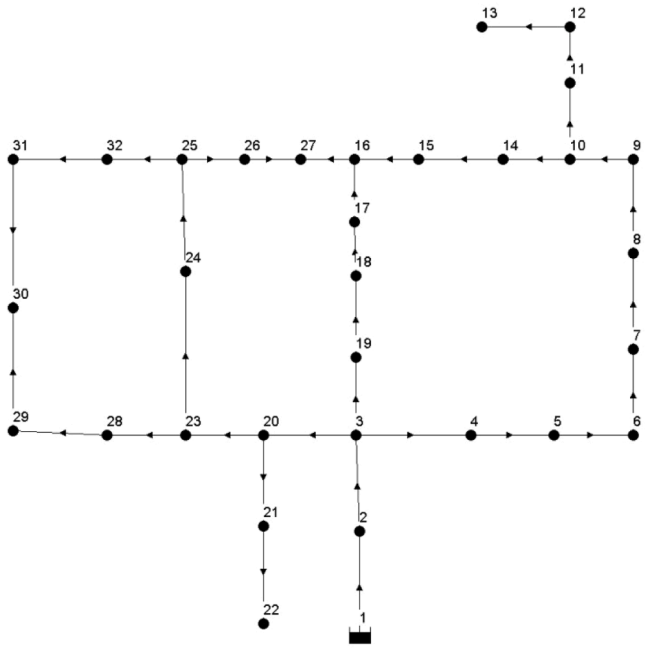

Fig. 2. topology of the DMA of the Hanoi WDN used in this paper.

random noise and, in the case of presence of leaks, also affected by their effects. This huge random noise consists of a unique random component applied to each aforementioned instantaneous withdrawal sample value at each WDN junction and is calculated as a normal distribution, with a standard deviation equal to **0.33** multiplied by the corresponding nominal instantaneous withdrawal value, around this nominal water demand. The pressures, from scenario to scenario, are also affected by variations in the morphology of the WDN (as a matter of fact, during the generation of synthetic data we used, the length, the diameter, and the roughness of the pipes could vary randomly by up to **25 %** of their nominal values) and, within each scenario, the pressures are also affected by corresponding variations in instantaneous withdrawals from the nodes of the WDN, including instantaneous noise, hourly variations, daily variations over the week and seasonal variations over the year. It is noteworthy that these variations are stochastically different for each scenario and, within each scenario, for each junction of the WDN. Moreover, the various pressure time series are not stationary stochastic processes and the pressures in the WDN do not depend linearly on the instantaneous withdrawals, on the WDN morphology and on leaks. In each scenario multiple leaks of different sizes could also be introduced randomly, and these leaks can either overlap temporally or to be distinct in time, and these leaks can also be of the 'abrupt' type, that is displaying random but fixed characteristics from its onset, or they can be 'incipient,' meaning they may increase in size from a value of zero to a random maximum value over a random time interval. Leaks, in our datasets, are always located at junctions in the WDN, have random durations and, from those two possible kinds of leaks, 'incipient' leaks are much more difficult to detect, because the pressure variations they cause blend with the random hourly, daily, and seasonal pressure variations that are typically present in the WDN even in the absence of leaks. Our method, in the case of multiple leaks in the same scenario, can initially identify only the most significant one; therefore, in a real case, to identify with our method any other further leak that may be present in the same scenario, it is first necessary to eliminate the detected leak and then to reapply the algorithm to the new pressure samples collected afterward. Unfortunately, in simulated data we used [22,23] it is not possible to eliminate an already detected leak and then to search for any other leaks present in the same scenario, because the pressure values that would have been present in the specific scenario at the specific

junctions in the absence of the leak to be eliminated are not known at all, and thus this issue prevents in our synthetic data the identification of multiple leaks.

The research group [22] that generated the pressure datasets also distributed a set of software libraries, named 'epyt' [29,30], which we also used, that allow a interaction, via programming with high level languages like Python (which we used), with the model created for the WDN using the Epanet [27,28] software.

2.2.3. *Pre-processing of pressure data before to feed them to the neural models*

For each scenario and for each junction of the WDN, the synthetic pressure data we used are organized into time series lasting one year and containing pressure values recorded every 30 min (see Figs 3 and 6), Each of these time series thus contains $48 \times 365 = \mathbf{17{,}520}$ pressure values. As previously mentioned, these series are affected by various types of stochastic noise and do not constitute stationary stochastic processes. Each time series related to each junction of the WDN changes, sometimes significantly, both in pressure instantaneous values and also in pressures values patterns, and this is caused by:

- the nonlinearity of the dependence of pressures in the WDN on withdrawals and on potential leaks;
- the daily periodic variations of withdrawals from the WDN;
- the weekly periodic variations of withdrawals from the WDN;
- the seasonal periodic variations of withdrawals from the WDN;
- the instantaneous stochastic variations of withdrawals from each junction in the WDN;
- the seasonal stochastic variations of withdrawals from each junction in the WDN;
- the scenario-to-scenario variations of these stochastic withdrawal variations;
- the random scenario-to-scenario variations in the morphology of the WDN;
- the random presence of potential leaks in the scenarios, whose characteristics (i.e., position, diameter, and type: 'abrupt' or 'incipient') also vary randomly from leak to leak;
- the random increase in the diameters of 'incipient' leaks over time;
- the random duration of the leaks,

Therefore, before to feed the different time series to neural models and in order to make these series each other comparable, a new type of data pre-processing is necessary and our new type of pre-processing for the time series of pressure data corresponding to a scenario and to a junction of the WDN follows these steps:

- low-pass filtering of this sequence using a Butterworth filter (see Figs 4,5 and 7). This filter, for the time series of data used to train the neural networks, has a cut-off frequency set at the coefficient number 20 of the Fourier transform of the time series (each Fourier coefficient corresponds to a multiple of a frequency of 1/1800 hz, that is, of 1/2 hour) and it has order 4, while, for the time series of data used to identify potential leaks in the WDN, it has a cut-off frequency set at the coefficient number 26 of the Fourier transform of the series and still has order 4. These cut-off frequencies have been chosen to eliminate from the pressure data sequences components with periodicities of variation less than or equal to about one week and have been refined through several tests with nearby cut-off frequencies;
- after filtering, the temporal delay of each output data sequence from the filter, respect to the corresponding input data sequence, is then eliminated;
- then, but mandatory and done only for the more complex 1000 scenarios dataset, from each filtered time series the average value of the series is subtracted;
- finally, in order to make the data from different junctions of the WDN comparable, the filtered pressure data undergo a normalization process which requires that, during the training phase of each Autoencoder, the individual sequences of filtered pressure data contained in the leakless scenarios used for the training are normalized separately as follows:

$$pk_{norm} = (pk - pk_{\min})/(pk_{\max} - pk_{\min}) \tag{2}$$

where $\mathbf{p_{k\_norm}}$ is the normalized and filtered pressure value at junction $\mathbf{k}$ of the WDN, $\mathbf{p_k}$ is the original filtered pressure value at junction $\mathbf{k}$, and $\mathbf{p_{k\_min}}$ and $\mathbf{p_{k\_max}}$ are the minimum and maximum values of $\mathbf{p_k}$ in the global sequence obtained by concatenating the individual filtered data sequences used for training the Autoencoder corresponding to junction $\mathbf{k}$. The values of $\mathbf{p_{k\_min}}$ and $\mathbf{p_{k\_max}}$ derived during the training phase are then stored for each junction $\mathbf{k}$ of the WDN and afterward used, during the inference phase of each trained Autoencoder, to normalize any other single sequence of pre-processed and filtered pressure data relating to the specific junction $\mathbf{k}$ they refer to.

2.2.4. *The 1D CNN neural network in an 'Overcomplete Autoencoder' configuration*

A 1D CNN neural network in an 'Overcomplete Autoencoder' configuration, as before mentioned, consists of a part named Encoder followed by and feeding a part named Decoder. As in our specific case, the Autoencoder, when receiving as input a sample consisting of a time series of pressure data values, is able to reconstruct this same sequence

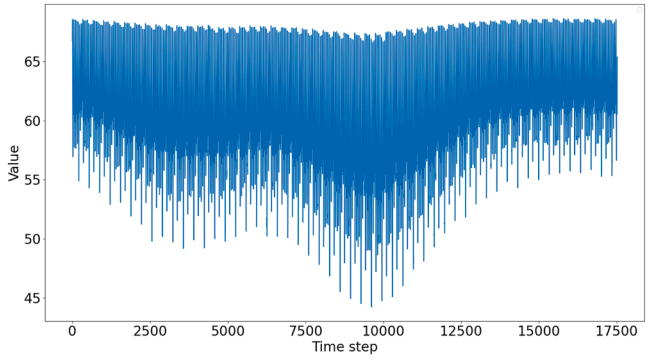

Fig. 3. original pressure data at junction 13 in scenario 425. There is a very small 'abrupt' leak present at this junction starting from pressure sample number 4445 which is not visible at all in the original data affected by a huge noise level. Notice the patterns in the pressure data with daily, weekly and seasonal periodicity.

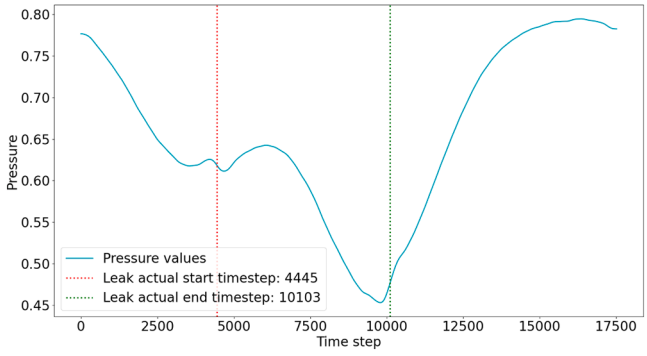

Fig. 4. pressure data at junction 13 in scenario 425 after low-pass filtering and normalization. The small 'abrupt' leak, starting from pressure sample number 4445, is more evident than in the original data. It is noteworthy that only seasonal patterns and patterns caused by the leaks remain in the filtered pressure data sequence.

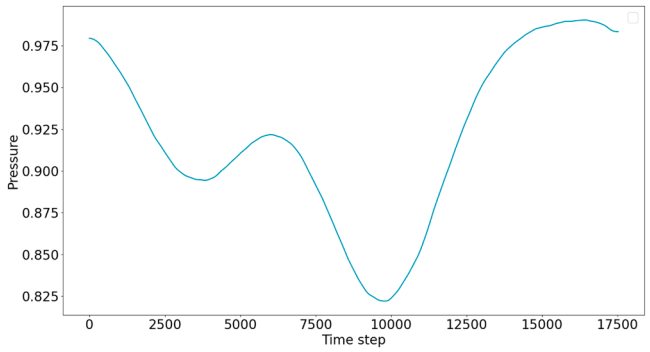

Fig. 5. pressure data at junction 13 in scenario 4 after low-pass filtering and normalization in the absence of leaks. Note also the difference, for the same junction examined before, in the pressures in this scenario compared to those in scenario 425 (see Fig. 4) even during periods in the latter where there are no leaks present. Again, it is noteworthy that only seasonal patterns remain in the filtered pressure data sequence.

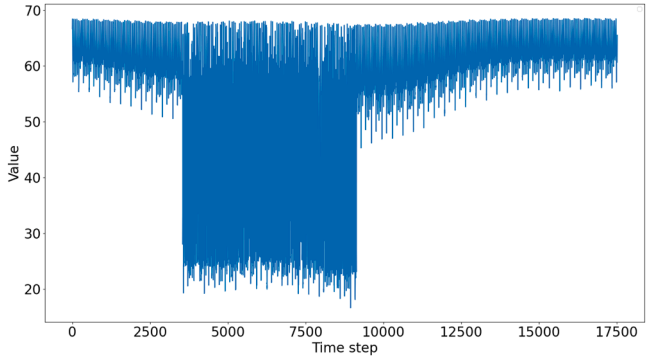

Fig. 6. original pressure data at junction 21 in scenario 5. There is a large 'incipient' leak present in this junction starting around sample number 3550 which is easily visible even in the original data. Again, note that there are also daily, weekly, and seasonal patterns present in the pressure data sequence.

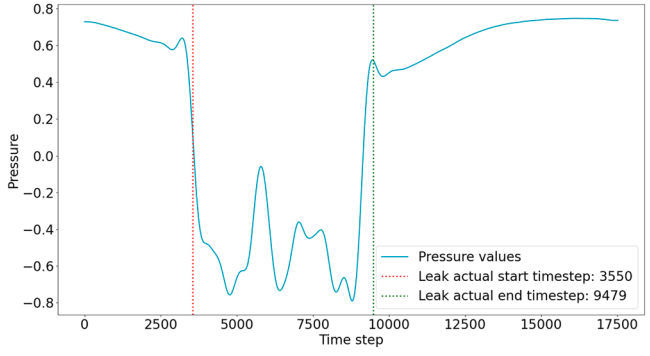

Fig. 7. pressure data at junction 21 in scenario 5 after low-pass filtering and normalization. There is a large 'incipient' leak at this junction starting from pressure sample number 3550. The glitches created by the low-pass filter applied to pressure data with very pronounced pattern variations are very evident in this graph. However, given the significant pattern variation of the original data, these glitches are still acceptable.

in output with limited errors, if and only if, this time series exhibits patterns similar to those of the time series previously used for training the Autoencoder, but generates much greater reconstruction errors when instead the input samples have sufficiently different patterns, even over a few time steps, from those of the samples used for training the Autoencoder. This reconstruction error caused by anomalies is further accentuated in an 'Overcomplete Autoencoder' like our ones, where the Encoders increases, instead of to decrease, the dimensionality of the input and where the Decoders reduces this dimensionality to restore it to its original value. In practice, with this mechanism, it is possible to detect anomalies in the time series and, as before mentioned, our work uses a dedicated 1D CNN neural network in 'Overcomplete AutoEncoder' configuration for each junction of the WDN.

The Encoder part of each 1D CNN we used is built in detail as follows:

1) a '*1D Convolutional Layer*' with 64 filters of size 7 and 'stride' 2, followed by a '*ReLU*' activation function;
2) three '*residual blocks*', each one consisting of two '*1D Convolutional Layers*', each one of these having 64 filters of size 7 and 'stride' 1, followed by a '*ReLU*' activation function. The output of the last 'ReLU' activation function of each residual block is added to the input of the block itself, and the sum is passed to the input of the next layer;
3) a '*1D Convolutional Layer*' with 32 filters of size 7 and 'stride' 2, followed by a '*ReLU*' activation function,

and the Decoder part, which is fed by the corresponding Encoder, of each 1D CNN we used, is built instead in detail as follows:

1) a '*1D Transpose Convolutional Layer*' with 32 filters of size 7 and 'stride' 2, followed by a 'ReLU' activation function;
2) three '*residual blocks*', each one consisting of two '*1D Transpose Convolutional Layers*', each one of these having 32 filters of size 7 and 'stride' 1, followed by a '*ReLU*' activation function. The output of the last 'ReLU' activation function of each residual block is added to the input of the block itself, and the sum is passed to the input of the next layer;
3) a '*1D Transpose Convolutional Layer*' with 64 filters of size 7 and 'stride' 2, followed by a 'ReLU' activation function;
4) a '*1D Transpose Convolutional Layer*' with 1 filter of size 8 and 'stride' 1, followed by a '*linear*' activation function.

Regarding all the used '*hyperparameters*' in our neural network, the '*batch size*' parameter is always set to 96, the '*padding*' is always of type 'same', and the '*dilation rate*' of the filters is always equal to 1. Furthermore, the number of '*epochs*' to train each neural network has been set equal to 200, a 'loss function' of type 'root_mean_squared_error' is always used, and an 'adam' '*optimizer*', with a '*learning rate*' of 0.001, is always employed. An '*Early Stopping Strategy*' was applied monitoring the '*loss function*' and using a '*patience*' of 10. A '*Reduce Learning Rate on Plateau*' of 0.2 was also implemented with a '*patience*' value of 5 and a '*Learning Rate Minimum*' of 0.0005. In the event of '*Early Stopping*', the weights that generated the minimum value of the '*Loss function*' are retained in the training neural model.

All the aforementioned '*hyperparameters*' were optimized with a 'search grid' phase and we found values for them similar to those previously determined by another group from the Politecnico di Milano for a 1D CNN neural network significantly different from ours. The training of each 1D CNN network was then conducted, without any validation, using two complete scenarios without leaks. In reality, more precisely, a CNN training on one scenario was followed by a CNN retraining on the other scenario.

2.2.5. Reconstruction errors of the 1D CNN in an 'Overcomplete Autoencoder' configuration

Each Autoencoder in our bank of neural networks (see Fig. 1), when used after its training to detect and localize leaks, generates reconstruction errors, which are the larger the more the pattern of the time series constituting the current input sample of the Autoencoder differs from the patterns of the time series used during the training phase of the same Autoencoder. At each moment **n** each Autoencoder in the Autoencoders bank receives a row of the matrix $\mathbf{I}^{\mathbf{n}}$ and consequently

generates a row of the matrix $\hat{\mathbf{I}}^{\mathbf{n}}$ with a corresponding reconstruction errors sequence.

To calculate the reconstruction error made by the Autoencoder bank at time **n** for each junction **k** of the WDN, the difference matrix **D** (see *Eq. (3)*) between the matrix $\hat{\mathbf{I}}^{\mathbf{n}}$ of current pre-processed pressure input data and the matrix $\mathbf{I}^{\mathbf{n}}$ of predicted pressure output data is then first constructed and, for each row of this difference matrix **D**, the root mean square value of the data contained in each row is calculated. The time series built with these root mean square values, one time series for each junction, constitute the sequence of reconstruction errors values used afterward for leaks detection and localization within the scenario under examination. In other words:

$$D = \hat{I}n - In \tag{3}$$

and, by analyzing, after a proper pre-processing, the patterns of these sequences of reconstruction errors, it is possible to identify and localize any leaks present in the WDN during the scenario being examined. For example, regarding junction 13, which, as noted above, has a very small 'abrupt' leak in scenario 425, and regarding junction 21, which instead has a significant 'incipient' leak in scenario 5, we can observe, after a proper pre-processing, the corresponding graphs of the corresponding reconstruction errors sequences generated by the respective Autoencoders (see Figs 8 and 10).

More specifically, the continuous green curve in Figs 8,9 and 10 is the result of a low-pass filtering, performed with a moving average filter having a rectangular window of 64 samples, on the time series of reconstruction errors made by the corresponding Autoencoder during the analysis of the scenario related to the specific graph at the indicated junction of the WDN and, from now on, the values represented by this green curve will be simply referred to as 'low pass filtered'.

The red curve in Figs 8,9 and 10 represents instead the result of low-pass filtering the data related to the green curve with an additional second-order Butterworth low-pass filter having a cut-off frequency set at the 7th coefficient of the Fourier transform of the data represented by the green curve and, from now on, the data related to the red curve will be simply referred to as 'low pass filtered'. The length of the window of the aforementioned moving average filter and the cut-off frequency of the Butterworth filter have been optimized through trials and are designed to attenuate both the signal components with a periodicity approximately less than or equal to the daily periodicity and the corresponding residual signal components with a periodicity approximately less than or equal to the weekly periodicity. After each filtering, a compensation for the delay introduced by the respective filter is always performed to make the various curves comparable.

The analysis of the pretreated reconstruction error time series committed during a scenario by an Autoencoder consists of searching for the maximum value of the difference between those corresponding two curves and this analysis is based on the idea that, by attenuating in the 'low pass filtered' error curve (the green one curve) all components with

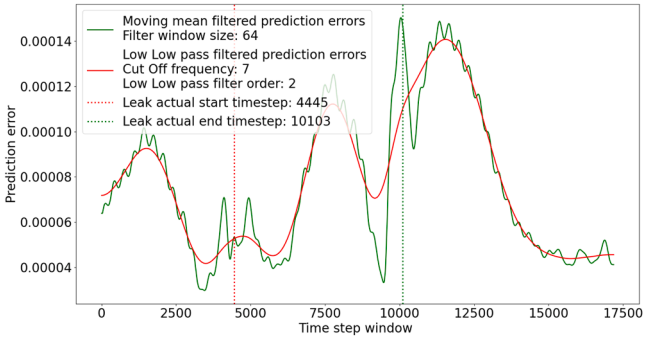

Fig. 8. Reconstruction errors sequence, after pre-processing, of the Autoencoder at junction 13 during the analysis of the scenario with leak 425.

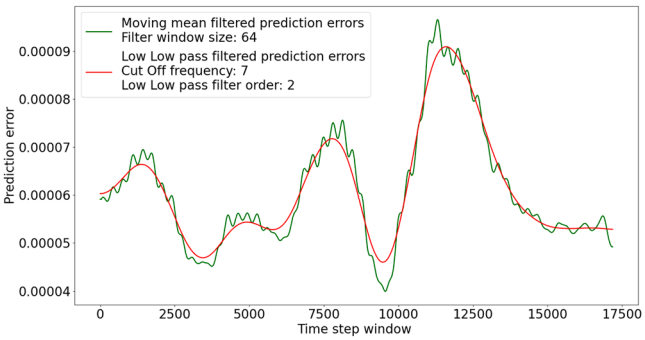

Fig. 9. Reconstruction errors sequence, after pre-processing, of the Autoencoder at junction 13 during the analysis of the leak-free scenario 4.

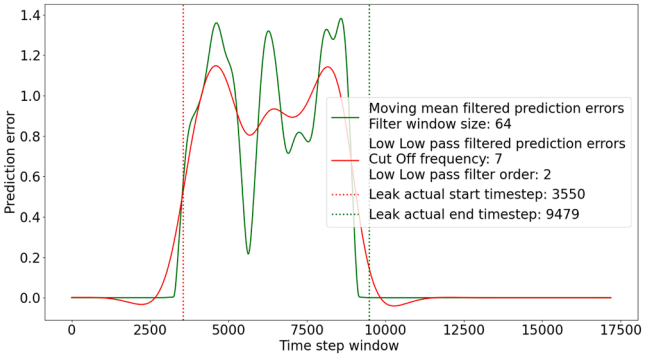

Fig. 10. Reconstruction errors sequence, after pre-processing, of the Autoencoder at junction 21 during the analysis of the scenario with leak 5.

a periodicity approximately less than or equal to the weekly periodicity, only components with an approximately seasonal periodicity remain in the 'low pass filtered' error curve (the red one curve) (see Fig. 9 regarding the 'low pass filtered' reconstruction errors committed for junction 13 during scenario 4 without leaks). As a matter of fact, it is observed that, in the case of scenarios without leaks, the aforementioned components with a periodicity approximately less than the weekly periodicity have smaller amplitudes than those found in scenarios with leaks. Indeed, as can be seen from the above figures, the presence of a leak at a junction in the WDN introduces both sudden variations and continuous variations in the pressure data patterns present in the WDN and also introduces high-frequency components in the error curves generated by the Autoencoders. These high-frequency components have a greater amplitude than those present in the absence of leaks and the amplitude of these high-frequency components is generally, but unfortunately not always, greater the larger the leak, and it is also generally larger in the case of 'abrupt' leaks compared to the case of 'incipient' leaks.

When, during a scenario, the maximum value of the aforementioned difference exceeds at instant **n**, in a junction of the WDN, a detection threshold we have set based on trials (good values for this threshold are **4.145E-5** for the 500 scenarios dataset and **7.488E-05** for the 1000 scenarios dataset), the scenario is considered to contain a leak and the leak is localized as being situated in the junction **k** of the WDN where this difference at instant **n** has the highest value. It is also possible to estimate the start time of the leak by choosing the instant **n** in the scenario under examination where the aforementioned difference first exceeds the detection threshold, or, more robustly but less precisely, by choosing the instant **n** in the scenario under examination where the aforementioned difference has the maximum value.

2.2.6. The identification of the distance between the junction where the leak is localized and a possibly different junction where the leak is actually present

Unfortunately, especially in the case of multiple leaks in the same scenario, with effects that may overlap in various junctions, or in the case of very small leaks, with effects not very dissimilar from noise, it is possible that the junction of the WDN where the algorithm localizes a leak is different from a junction of the WDN where the leak is actually present. To determine the distance in 'hops', i.e. in the number of junctions present between the junction of the WDN where the leak is localized and a possibly different junction of the WDN where the leak is actually present, and to determine the corresponding distance in meters, we decided to use the Dijkstra's 'Shortest Path' algorithm applied to the underlying graphs of the WDN. Each node of the WDN can indeed be seen as the vertex of a graph, and each pipeline of the WDN can be seen as an edge of this graph.

Using the aforementioned 'epyt' software libraries [29,30] we implemented a program written in Python that allows, querying the Epanet model of the WDN's DMA of Hanoi and extracting in such a way the necessary WDN-related data, to automatically construct the adjacency matrix of the underlying graph of the WDN. To determine that distance in 'hops', we considered an undirected and unweighted graph (i.e. with a length of 1 for each edge) and we applied Dijkstra's 'Shortest Path' algorithm [31] to its adjacency matrix. In such a way we determined the minimum distance in 'hops' between the junction of the WDN where the leak is localized and a possibly different junction of the WDN where the leak is actually present (we know this junction because it is specified within the synthetic dataset). To determine the minimum distance in meters between these junctions, we considered instead the undirected but weighted graph underlying the WDN built assigning at each 'edge' of this graph a weight equal to the length in meters of the corresponding pipeline of the WDN. Again, we obtained the length of each pipeline of the WDN by querying the Epanet model of the WDN, and again, knowing all these lengths, we constructed the corresponding weighted adjacency matrix of the graph. By applying Dijkstra's 'Shortest Path' algorithm [31] to this new adjacency matrix we finally determined the minimum distance in meters between the junction of the WDN where the leak is localized and a possibly different junction of the WDN where the leak is actually present.

The distances in 'hops' are shown in Fig. 11:

3. Results

3.1. Leak detection results

Unfortunately, until now, we have not been able to obtain real data from a real WDN, but, by applying the methodologies described above on aforementioned **500** scenarios dataset [22,23] (380 with leaks and 120 without leaks), we achieved the same some very good results (see Tables 1,2 and 3) with an **accuracy** of **92.5 %** corresponding to a true positive rate $\mathbf{R_{tp}}$ of **90 %**, a false positive rate $\mathbf{R_{fp}}$ of **0 %** and a true negative rate $\mathbf{R_{tn}}$ of **100 %**. The same methodologies applied for validation to **1000** scenarios (762 with leaks and 238 without leaks) of the aforementioned **different** and **more complex** validation dataset with smaller leaks, constituted by **1000** scenarios [22,23], then showed off a still good accuracy value of **87 %**. In comparison, another research group [16] of Politecnico di Milano reported to have achieved, on aforementioned **500** scenarios dataset, an **accuracy** of **89 %**, which is lower than ours of **92.5 %**, with a study conducted on the same our data, but using a different 1D CNN neural network, different data pre-processing and different CNN errors processing. Furthermore, the MNF-based leak detection methods, developed by the researchers who generated the aforementioned **500** scenarios dataset we used, obtained, at best, on these same data, a true positive rate $\mathbf{R_{tp}}$ of **68.98 %** with a true negative rate $\mathbf{R_{tn}}$ of **66.96 %** [22]. These values correspond to an **accuracy** of about **68.5 %** and again these values are worse than our **accuracy** of **92.5 %**, our true positive rate $\mathbf{R_{tp}}$ of **90 %** and our corresponding true negative rate $\mathbf{R_{tn}}$ of **100 %**.

Other researchers [19] have processed the same data using both RNN GRU and RNN LSTM neural networks. However, with these neural networks, they were only able to detect leaks greater than **80** l/s, whereas the smallest leak we accurately detected at the junction where it actually occurred was smaller than **50.4** l/s. The **accuracy** of their results is around **48 %**, with a true positive rate $\mathbf{R_{tp}}$ of **39.01 %** and a true negative rate $\mathbf{R_{tn}}$ of **74.96 %**, all of which are worse than ours corresponding results.

Regarding the distance in 'hops' or in meters between the junction of the WDN where the leak is located and the junction where the leak is actually present, we obtained an average distance in 'hops' of **1.23** 'hops' (see the histogram in Figure 11) and a corresponding average distance of **1219.8** m. In comparison, in its work, the above other

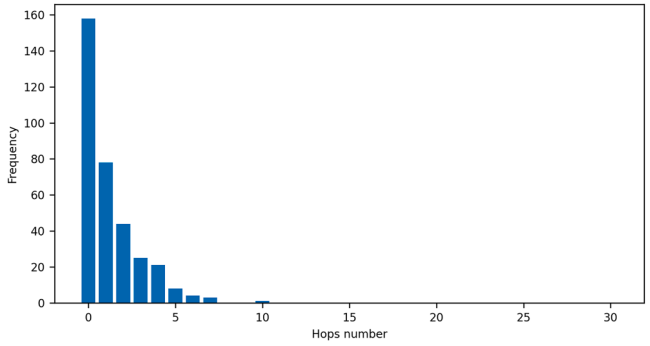

Fig. 11. Histogram of the distances in 'hops' between the junction of the WDN where a leak is localized and a possibly different junction of the WDN where the leak is actually present.

Table 1
A comparison of accuracy obtained in leaks detection using the same our WDN and same our data, but applying different methods.

Citation	*Leak detection method on the **500** scenarios dataset*	*True positives rate (Rtp)*	*False positives rate (Rfp)*	*Accuracy*
	Our method	~ **90 %**	**0 %**	~ **92.5 %**
[16]	CNN based method of other research group of Politecnico di Milano			89 %
[22]	MNF bases method of researchers who generated the used datasets	68.98 %	33.04	~ 68.5 %
[19]	RNN based methods of NTNU researchers	39.01 %	25.04 %	~ 48 %

Table 2
A comparison of Rtp and Rfp results obtained in leaks detection using a WDN and data different from ours and applying other methods (Martin Bjerke [19] and Chan et al. [20]). The citations [..] inside the above table are the citations in Chan et al. paper [20]. A.N.N. stands for "Artificial Neural Network", B.I.S. stands for "Bayesian Inference System", S.P.C. stands for "Statistical Process Control" and S.V.M. stands for "Support Vector Machine".

Citation	*Leak detection method*	*Data type*	*True positives rate (Rtp)*	*False positives rate (Rfp)*
[68]	Non-linear Kalman Filter	simulated data	87 %	0.01 %
[57]	A.N.N., S.P.C., B.I.S.	engineered test/ historical data	80 %/76 %	10 %/8 %
[27]	Ensemble CNN-S.V.M.	engineered test	98.2 %	0.2 %
[77]	Multiclass S.V.M.	simulated data/ historical data	99.5 %/98.75 %	

Table 3
A comparison of Rtp and Rfp results obtained in leaks detection using a WDN and data different from ours and applying other methods (Martin Bjerke [19] and Wu, Liu [21]). T.D.N.N stands for Time Delay Neural Network, M.D.N. stands for "Misture Density Network", F.I.S. stands for "Fuzzy inference system", A.N.N. stands for "Artificial Neural Network", B.I.S. stands for "Bayesian Inference System"", S.P.C. stands for "Statistical Process Control" and S.V.M. stands for "Support Vector Machine".

Leak detection method	*Data type*	*True positives rate (Rtp)*	*False positives rate (Rfp)*
A.N.N., T.D.N.N.	historical data	75 %	0 %
M.D.N. and F.I.S.	historical data	100 %	15 %
A.N.N., S.P.C., B.I.S.	historical data	100 %	8 %
Modified S.V.M.	historical data	80 %	10 %

research group [16] of Politecnico di Milano reported a mean distance of **1.78** 'hops' between the junction of the WDN where the leak is located and a possibly different junction where the leak is actually present.

4. Conclusions

This paper describes a new fully Data-Driven method that, by a new type of 1D CNN exploiting an Overcomplete Autoencoder configuration (Overcomplete AE), by a new method for data pre-processing and by a new type of analysis of the reconstruction errors time series committed by the Autoencoders, allowed us to detect and localize leaks within the scenarios contained in the 'LeakDB' standard benchmark synthetic dataset of 500 scenarios related to a DMA of the Hanoi WDN. We achieved an **accuracy** of **92.5 %** and an average distance in 'hops' of **1.23** 'hops' between the junction where the leak is located and a possibly different junction where the leak is actually present. Thus, this method has proven to be very effective compared to the state of the art in leak detection and localization methods applied to the same synthetic data (see Table 1) and also with respect to other methods applied on other kind of data (see Tables 2 and 3).

Unfortunately until now we have not been able to obtain real data from a real WDN, but, as soon as possible, we also intend to apply this method to pressure data collected in-field, specifically to pressure data obtained from pressure sensors installed in a real WDN and, with real data, the results for leak detection and localization could be even better, because, as argued by some researchers [19], the morphology of the part of a real WDN included in a predefined set of junctions changes very slowly and gradually over time; not suddenly, with variations of up to **25 %** from one year to the next and from one scenario to another, as is the case of the synthetic datasets we used. As a matter of fact that current literature [19] considers appropriate to generate new synthetic data using the same algorithm employed to create the data we used, but preemptively eliminating or mitigating these variations in WDN morphology and correcting also other unrealistic flaws present in the data we used that hinder the detection of 'incipient' leaks (which are the hardest to detect).

Nevertheless, in the presence of small or medium leaks in the DMA of the Hanoi WDN, even when applied to the regenerated and corrected synthetic data, the leak detection methods based on recurrent neural networks of type RNN LSTM and RNN GRU proposed by those researchers [19] did not provide results they deemed acceptable, and this fact further confirms the validity of the work we presented whose results instead pertain large, medium and even small leaks.

The heavy low pass filtering applied to pressures time series during the pre-preprocessing phase has proven to be able to eliminate almost any issue caused by huge instantaneous noise added to pressures data and thus our method has proven to be robust against this noise, but unfortunately our method has instead proven to be more prone to errors caused by seasonal random variations of withdrawals from each junction in the WDN, because each junction, also in absence of any leak, has in general a seasonal pressures dynamics different from other junctions and some pressures dynamics can completely overlap and hide the pressures dynamics introduced by small leaks. To address this issue, we are testing some new normalizing methods aimed to equalize and to minimize all the pressures dynamics that junctions could have in absence of leaks.

The pressures in used datasets, from scenario to scenario, are also affected by variations in the morphology of the WDN (as a matter of fact, during the generation of synthetic data we used, the length, the diameter, and the roughness of the pipes could vary randomly by up to **25 %** of their nominal values), but our method has proven to robust also against this issue.

CREDIT author statement

Elvio Damonti: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization

Giancarlo Bernasconi: Conceptualization, Validation, Resources, Writing - Review & Editing, Supervision, Project administration

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.dsp.2025.105603.

Data availability

Data will be made available on request.

References

[1] S. Mounce, J. Boxall, J. Machell, Development and verification of an online artificial intelligence system for detection of bursts and other abnormal flows, ASCE J. Water Resour. Plan. Manag. (2010) 309–318.
[2] S. Sarkamaryan, A. Haghighi, A. Adib, Leakage detection and calibration of pipe networks by the inverse transient analysis modified by gaussian functions for leakage simulation, J. Water Supply: Res. Technol.-Aqua (2018) 404–413.
[3] G. Guo, X. Yu, S. Liu, Z. Ma, Y. Wu, X. Xu, X. Wang, K. Smith, X. Wu, Leakage detection in water distribution systems based on time–Frequency convolutional neural network, ASCE J. Water Resour. Plan. (2021) 1–11.
[4] C. Karney, Energy and costs of leaky pipes: toward comprehensive picture, ASCE J. Water Resour. Plan. Manag. (2002) 441–450.
[5] J. Mashford, D.D. Silva, S. Burn, D. Marney, Leak detection in simulated water pipe networks using svm, Appl. Artif. Intell. (2012) 429–444.
[6] R. Perez, G. Sanz, V. Puig, J. Quevedo, M.A. Cuguero Escofet, F. Nejjari, J. Meseguer, G. Cembrano, J.M. Mirats Tur, R. Sarrate, Leak localization in water networks: a model based methodology using pressure sensors applied to a real network in Barcelona, IEEE Control Syst. Mag. (2014) 24–36.
[7] S. Adachi, S. Takahashi, H. Kurisu, H. Tadokoro, Estimating area leakage in water networks based on hydraulic model and asset information, in: Water Distribution System Analysis Conference, 2014. WDSA.
[8] N. Mashhadi, I. Shahrour, N. Attoue, J. El Khattabi, A. Aljer, Use of machine learning for leak detection and localization in water distribution systems, Smart Cities (2021) 1293–1315.
[9] P. Irofti, F. Stoican, V. Puig, Fault handling in large water networks with online dictionary learning, J. Process. Control (2020) 46–57.
[10] T.K. Chan, C.S. Chin, X. Zhong, Review of current technologies and proposed intelligent methodologies for water distributed network leakage detection, IEEe Access. (2018) 78846–78867.
[11] M. Romano, Z. Kapelan, D. Savi, Automated detection of pipe bursts and other events in water distribution systems, ASCE J. Water Resour. Plan. Manag. (2014) 457–467.

[12] Leak detection and localization in water distribution networks by combining expert knowledge and data-driven models, in: Adrià Soldevila, Giacomo Boracchi, Manuel Roveri, Sebastian Tornil-Sin, Vicenç Puig (Eds.), Leak detection and localization in water distribution networks by combining expert knowledge and data-driven models, Neural Comput. Appl. 34 (2022) 4759–4779.
[13] D. Loureiro, C. Amado, A. Martins, D. Vitorino, A. Mamade, S.T. Coelho, Water distribution systems flow monitoring and anomalous event detection: a practical approach, Urban Water J. (2016) 242–252.
[14] E. Galdiero, F.D. Paola, N. Fontana, M. Giugni, D.A. Savic, Decision support system for the optimal design of district metered areas, J. Hydroinformat. (2016) 49–61.
[15] A method based on VMD improved by SSA for leak location of water distribution, in: Zhi Yu a, Bo Tang a b, Wei Chen b, Danguang Huang b, Lei Xu b. s.l (Eds.), A method based on VMD improved by SSA for leak location of water distribution, Digit. Signal. Process. 145 (2024).
[16] Daniele Ugo Leonzio, Paolo Bestagini, Marco Marcon, Gian Paolo Quarta, Stefano Tubaro, Water leak detection and localization using convolutional autoencoders, in: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing, 2023. ICASSP.
[17] Leakage detection in water distribution networks via 1D CNN deep autoencoder for multivariate SCADA data, in: Hoese Michel Tornyeviadzi, Razak Seidu (Eds.), Leakage detection in water distribution networks via 1D CNN deep autoencoder for multivariate SCADA data, Eng. Appl. Artif. Intell. 122 (2023).
[18] Hridik Punukollu, Vasan Arunachalam, K. Srinivasa Raju, Leak detection in water distribution networks using deep learning, ISH J. Hydraul. Eng. (2022) 1–9.
[19] NTNU, Martin Bjerke. Leak detection in water distribution networks using gated recurrent neural networks. 2019.
[20] T.K. Chan, C.S. Chin, X. Zhong, Review of current technologies and proposed intelligent methodologies for water distributed network leakage detection, IEEe Access. 6 (2018) 78846–78867.
[21] Yipeng Wu, Shuming Liu, A review of data-driven approaches for burst detection in water, Urban Water J. 14 (9) (2017) 972–983.
[22] Vrachimis, S.G., Kyriakou, M.S., Eliades, D.G., and Polycarpou, M.M. LeakDB: a benchmark dataset for leakage diagnosis in water distribution networks. 2018. Proceeding of the WDSA /CCWI Joint Conference (Vol. 1).
[23] https://github.com/KIOS-Research/LeakDB.
[24] D. Bank, N. Koenigstein, and R. Giryes. Autoencoders. 2020, Arxiv.
[25] P. Bestagini, F. Lombardi, M. Lualdi, F. Picetti, S. Tubaro, Landmine detection using autoencoders on multipolarization GPR volumetric data, IEEE Trans. Geosci. Rem. Sens. (2021) 182–195.
[26] S.K. Yarlagadda, D. Güera, P. Bestagini, F.M. Zhu, S. Tubaro, E.J. Delp, Satellite image forgery detection and localization using GAN and one-class classifi er, IS&T Electron. Imaging (2018). EI.
[27] https://www.epa.gov/water-research/epanet.
[28] Rossman, L. “Epanet 2.0 user manual. 2000.
[29] Kyriakou, M.S., Demetriades, M., Vrachimis, S.G., Eliades, D.G., .& Polycarpou, M. M. https://github.com/OpenWaterAnalytics/EPyT.
[30] M.S. Kyriakou, M. Demetriades, S.G. Vrachimis, D.G. Eliades, M.M. Polycarpou, EPyT: an EPANET-python toolkit for smart water network simulations, J. Open. Source Softw. (2023).
[31] Dijkstra, W. Edsger, A note on two problems in connexion with graphs, Numer. Math. (Heidelb) (1959) 269–271.