Neurocomputing 438 (2021) 195–208

Contents lists available at ScienceDirect

Neurocomputing

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

Robust leak localization in water distribution networks using computational intelligence

Marcos Quiñones-Grueiro [a], Marlon Ares Milián [a], Maibeth Sánchez Rivero [a], Antônio J. Silva Neto [b], Orestes Llanes-Santiago [a,*]

[a] Universidad Tecnológica de La Habana José Antonio Echeverría, Calle 114 No. 11901, CUJAE, Marianao, La Habana CP:19930, Cuba
[b] Instituto Politécnico-Universidade do Estado do Rio de Janeiro, Rua Bonfim No. 25 – Vila Amelia, Nova Friburgo, RJ CEP: 28625-570, Brazil

ARTICLE INFO

Article history:
Received 15 September 2019
Revised 18 February 2020
Accepted 2 April 2020
Available online 26 January 2021

Keywords:
Computational intelligence
Deep learning
Gaussian process
Topological differential evolution
Temporal analysis
Robust leak location
Water distribution networks

ABSTRACT

The search for new strategies for leak detection, estimation and localization in Water Distributions Networks (WDNs) is a state-of-the-art research topic. In this paper, a methodology for leak detection, estimation and location that combines data-driven and model-based methods is proposed. A deep neural network is used in the leak detection task. Subsequently, the estimation of a leakage size range is accomplished by using Gaussian process regression. Then, a novel approach based on the solution of an inverse problem is developed for leak location. Knowing the range of possible values for the leak size allows to improve the location task when solved as an inverse problem. The proposed location method considers the topological configuration of the network as well as the leak size range. One of the main advantages of the proposal is that it does not depend on the labeling of the nodes. In this sense, a modified variant of the Differential Evolution algorithm, which considers the topological structure of the WDN to modify the search space and incorporates a temporal analysis, is used to find the solution of the inverse problem. Moreover, thanks to the topological evolution of the solutions a set of candidate nodes for the leakage creates a zone of reduced possible locations very useful in practical terms. The proposed approach is tested with the model of a real case study: the large-scale Modena WDN. The results demonstrate the effectiveness of the proposal with satisfactory leak detection, leak size estimation, and location performance when considering only 9 sensors installed in a network formed by 268 nodes.

1. Introduction

Currently, more than half of the world's population live in urban areas and this quantity will continue to increase in the upcoming years [43]. An essential element for life is water. Therefore, the increase of the population in the urban areas requires a more efficient management of the water resources. Water distribution networks (WDNs) are used to distribute drinking water to customers. One of the aspects that affects the efficient use of water resources are leaks in WDNs. Therefore, leakage detection and location is a fundamental task to guarantee their efficient operation.

For a better management of water resources, the International Water Association (IWA) proposed the principle of district metering. Under this principle, the distribution system is subdivided into discrete zones or district meter areas (DMAs) which permit the permanent closure of valves, with the idea of allowing the characterization of the normal flow and pressure variations [18].

On the other hand, it is known that the use of methods of analysis of measurements obtained from DMAs allows to develop proactive strategies in the detection and location of leaks. These strategies are divided in two groups: direct observation and inference methods [7]. Direct methods are based on different physical principles fundamentally related with the areas of acoustic and vibrations. They employ highly specialized hardware as for example highly sensitive acoustic detectors, hydrophones and accelerometers. This equipment is expensive and trained and specialized personnel is required to use it [11,21,23].

In the case of the inference methods, it is possible to build models for representing the DMA behavior by using the data obtained from permanently installed sensors. The most used inference strategy for leak detection in WDNs is the water balance method under minimum night flow regime [10]. This approach requires the installation of flow meters in some pipes of the network for collecting flow data. However, water companies usually measure flow only in the input and output of each DMA. Moreover, the

* Corresponding author.
E-mail addresses: marcosqg@gmail.com (M. Quiñones-Grueiro), mjares1997@gmail.com (M. Ares Milián), maibethsanchez91@gmail.com (M. Sánchez Rivero), ajsneto@iprj.uerj.br (A.J. Silva Neto), orestes@tesla.cujae.edu.cu (O. Llanes-Santiago).

https://doi.org/10.1016/j.neucom.2020.04.159
0925-2312/

installation and maintenance of these sensors is expensive and time-consuming. An alternative solution adopted by water companies is to install pressure head sensors in some nodes of the network because their price is lower than flow meters as well as they are easier to install and maintain.

In the scientific literature, there are several residual-based analysis methods proposed for leakage detection and location. A residual signal vector represents the difference between the outputs of a hydraulic model and the measurements obtained from the sensors installed in the network. A number of approaches use an analytic model of the DMA based on first principles to simulate the operation scenarios and generate then explicitly the residual vectors. The methods that follow this approach are referred as model-based methods in this paper. The key for successful leak diagnosis of this approach is to accomplish a satisfactory calibration of the model as well as an accurate characterization of the demand patterns [35]. Therefore, the application of the analytic-based methods is limited by the model's uncertainties.

Data-driven approaches use pattern recognition and statistic-based models for leak diagnosis constructed from historical data obtained from the DMA [1,22],25,46. This approach has achieved satisfactory results in the leak detection task because, in general, sufficient data of the normal behavior of the DMA is available. In addition, it does not requires detailed knowledge of the network parameters [15]. The main limitation of these approaches is in the location process because data from leakage scenarios are generally scarce [45].

With the objective to use the advantages of the both of the above groups of strategies, a mixed analytic/data-based approach has been developed for improving the leakage location process [37,38]. These papers propose the combination of a calibrated hydraulic model of the DMA with a classification method for leak location. The main advantages of this approach is the capacity of the data-based methods to handle uncertainties, and the possibility to train the classifier with synthetic scenarios that can be generated with the hydraulic model. The achieved results show better performances with respect to model-based methods such as those based on the leak signature matrix (also known as sensitivity matrix) [27,28].

Despite the high and diverse amount of proposed methods for detecting, estimating and locating leaks in water distribution networks, this topic continues to be an area of intense research. Some of the factors that motivate this interest are: a large number of the proposed solutions do not take into account the uncertainties caused by a varying water demand of the consumers, some proposal need a high number of sensors that have a high cost, and the performance that has been achieved demands the development of new more effective strategies.

The two main contributions of this paper are: 1) a new strategy for leak detection, estimation and location that combines data driven and model-based methods; and 2) a variant of Differential Evolution Algorithm named T-DE-TA (Topological Differential Evolution with Temporal Analysis). With the objective of achieving a high performance in the detection task a deep neural network is used. Subsequently, the estimation of a leakage size range is made by using Gaussian process regression because of the knowledge of the possible values for the leak size allows to improve the location task. Then, a novel approach based on the solution of an inverse problem is presented for leak location. The proposed location strategy considers the topological configuration of the network and the leak size range. One of it's main advantages is that it does not depend on the labeling of the nodes. In this sense, a modified variant of the Topological Differential Evolution algorithm presented in [34], which incorporates a temporal analysis, is used in the solution of the inverse problem.

The structure of the article is the following. Section 2 presents the general characteristics of WDNs. In Section 3, the new methodology and the computational tools used for leak detection, estimation and location are presented. In Section 4, the case study of the Modena WDN is presented. In Section 5, the results obtained with the application of the proposed methodology to the case study are analyzed. Finally, conclusions and suggestions of future research are presented.

2. Features of WDNs

A WDN is formed by P nodes and Q junctions. WDNs are fed by one or several reservoirs to provide water to different types of users. To guarantee a minimum pressure head in every node of the network, the installation is designed based on the use of pumps or gravity-fed. In this paper, the latter case is considered. Therefore, the behavior of the network depends on the consumers and the physical laws that govern the system. The main first principle laws that describe the network are.

- Conservation of mass: the net inflow must be equal to the net outflow for any node $p \in P$ of the network:

$$\sum_{j=1}^{b_p} q_j = d_p \tag{1}$$

  where b_p represent the number of branches connected to the node p, q_j denotes the flow of the branch j and d_p is the demand in this node.
- Conservation of energy: The sum of pressure heads around any loop of the network is equal to zero. If a loop of the network has S water sources and D water drops, it is modeled by

$$\sum_{s=1}^{S} h_s + \sum_{d=1}^{D} h_d = 0 \tag{2}$$

  where h_s represents the pressure heads in the sources and h_d represents the pressure heads in the drops.
- Flow-pressure relationship: The relation between flow q and pressure head h for any component in the network is

$$h = \theta q^{\gamma} \tag{3}$$

  where the parameter θ depends on the specific component and γ could have a value close to 2 [13].

The model of a WDN is mainly represented by algebraic nonlinear equations which are solved for steady-state conditions. In general, the dynamic behavior is not considered because the sampling time of real installations varies from 15 to 60 min. Thus, the dynamics are not properly captured and most analysis for operation management (i.e. leak detection, estimation and location) consider steady state conditions.

2.1. Water leaks

In general, leak detection and localization methods for water distribution networks assume the occurrence of leaks only in nodes of the network, where they can be described as an extra demand or a demand dependent of the pressure [28]. Even if leakages can occur at any point of the network, this simplification is considered in most works to facilitate the simulations [5,8].

For a realistic approach to simulate how leaks affect the behavior of WDNs, the following model of a leak with magnitude l_p occurring in node p is considered

$$l_p = (C_e) h_p^{\gamma} \tag{4}$$

where C_e represents an emitter coefficient that is proportional to the size of the leakage, h_p is the pressure head at the node, and $\gamma = 0.5$ [32].

3. Methodology for leak detection, estimation and location

The methodology proposed in this paper, which is presented in Fig. 1, is composed by three fundamental stages: leak detection, leak estimation and leak location which together form an integral strategy.

Obtaining a high performance in the detection stage is essential for the integral result of the proposed methodology since the other two stages depend on these results. In this stage a deep neural network is used to evaluate the residue obtained from the difference between the output measurement vector of the WDN and the output measurement vector estimated by the WDN model in the Hydraulic Simulator. If a leak is detected, the leak estimation stage is activated.

The estimation task is solved by using a Gaussian process regression method which uses the residue to estimate the leak range. This information is very important in the location task because it allows to reduce the search area by improving the location results.

Finally, in the location stage, an inverse problem approach solved as an optimization problem is developed. For solving the optimization problem a metaheuristic algorithm adapted to operate in WDNs is used.

Following, each one of the stages and methods are presented in details.

3.1. Leak detection

The solution of the leak detection task with data-driven methods can be formalized as a binary classification problem as follows.

Definition 3.1 (*Data-driven leak detection task*). Given a data set $D = \{\mathbf{x}\}_{i=1}^{nc}$ ($\mathbf{x} \in \Re^m$) of *nc* observations acquired during normal conditions that is representative of the nominal operation of the WDN, and a data set $D_l = \{\mathbf{x}\}_{j=1}^{lc}$ of *lc* observations acquired during leak conditions that is representative of the leaks to be detected in the WDN, find a function $f_d(\mathbf{x}) : \Re^m \rightarrow [0, 1]$ that maps any observation of the feature space to a set of two possible scenarios 0: normal and 1: fault.

As it is shown in Fig. 1 the feature vector $\mathbf{x}$ is formed by the residuals $\mathbf{r}$ obtained from the difference between the variables measured in the real WDN and the variables estimated from the WDN model.

3.1.1. Deep neural networks

In order to detect the presence of leaks in the WDN a deep neural network (DNN) will be used. The motivation for using DNNs as leak detection methods is twofold: first, DNNs allow to model nonlinear relationships among the variables which is a characteristic feature of WDNs, second, DNNs have demonstrated to be powerful feature extractors with different levels of abstraction such that the important features for the leak detection task can be captured.

Neural networks are models whose parameters are learned with continuous optimization methods. Neural networks are inspired by the way the brain works. The biological mechanism is simulated through the artificial neural network basic building block: the neuron. This block can be represented mathematically as follows

$$f(\mathbf{x}) = y = \sigma(w_0 b + w_1 x_1 + w_2 x_2 + \ldots + w_n x_N) \tag{5}$$

where y is the output of the neuron, $\mathbf{x} = [x_1, \ldots, x_N] \in \Re^N$ constitutes a vector of neuron inputs, w_j represents the weight corresponding to input x_j (with $j = \{1, 2, \ldots, N\}$ variables), σ is the called activation function, and $w_0 b$ is a bias (for this paper: $b = 1$). A layer in an artificial neural network is formed by a set of neurons which share the same inputs [6].

A deep neural network with L layers is defined as a composition of L functions $f_i : E_i \times H_i \rightarrow E_{i+1}$, where E_i, H_i and E_{i+1} are inner product spaces for all $i \in L$. The vector $\mathbf{x}_i \in E_i$ represent the input to layer L_i and the vector $\mathbf{w}_i \in H_i$ represent the weights or *parameters* of layer L_i [6]. The output of the network is then calculated by

$$F(\mathbf{X}; W) = (f_L \circ \ldots \circ f_1)(\mathbf{X}) \tag{6}$$

where $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_L\}, W = \{\mathbf{w}_1, \ldots, \mathbf{w}_L\}$ and $\circ$ represents the convolution mathematical operation.

DNNs have become very popular dealing with a wide range of tasks such as speech recognition, computer vision, natural language processing, and others [16]. A reduction in depth (amount of layers) results in an exponential sacrifice in width (amount of neurons per layer), increasing the amount of parameters needed to achieve a good approximation, which results in a higher computational cost [19]. This fact motivates the use of a DNN in this paper.

The parameters of a neural network are adjusted by finding the values of the weights that minimize a cost function C by using the backpropagation algorithm. The updating step of the weights from a single layer can be defined as follows [36]:

$$\mathbf{w}_{updated} = \mathbf{w}_{old} - \eta \bigtriangledown C \tag{7}$$

where $\mathbf{w}$ represents the vector of weights, $\bigtriangledown C$ represents the gradient of the cost function, and η the learning rate.

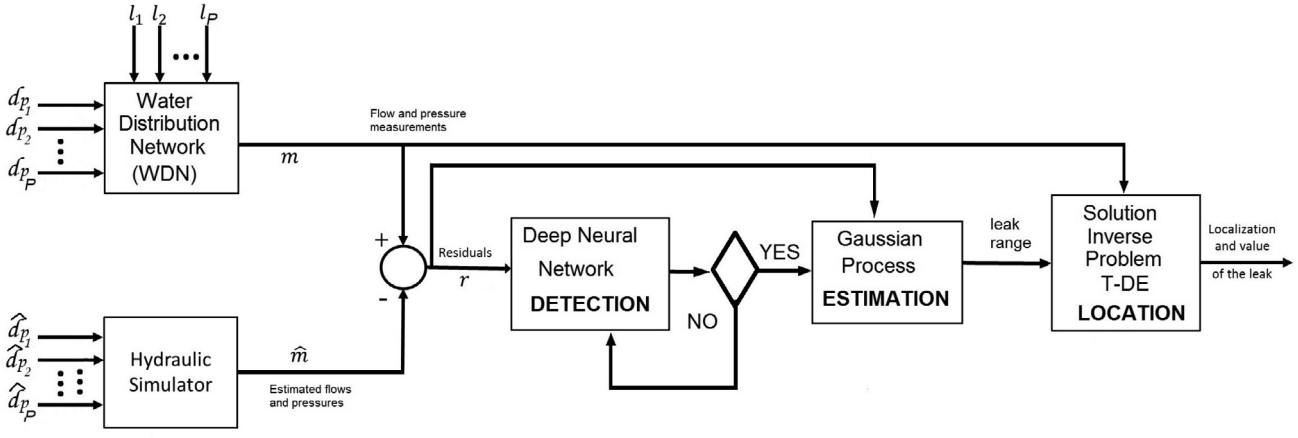


Fig. 1. Methodology proposed for detection, estimation and localization of leaks in a WDN.

3.1.2. Proposed artificial neural network

The neural network architecture shown in Fig. 2 consists of an input layer with as many measurement sensors as network input variables. This input layer is connected to a set of hidden layers. The amount of hidden layers and the amount of nodes per hidden layer define the architecture of the network. The activation function connecting these layers (input-hidden and hidden–hidden) is the Rectified Linear Unit function:

$$\varphi(z) = \begin{cases} z & \text{if} \quad z \geqslant 0 \\ 0 & \text{if} \quad z < 0 \end{cases} \tag{8}$$

where z corresponds to the input variable.

Following the hidden layer(s) is the output layer with one output signal $f(o)$. The activation function for this output is the sigmoid function:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{9}$$

The output function $f(o)$ is defined as (10)

$$f(o) = \begin{cases} 1\,(leak) & \text{if} \quad o \geqslant \mu \\ 0\,(normal) & \text{if} \quad o < \mu \end{cases} \tag{10}$$

where μ is a threshold parameter to be adjusted in the network.

In order to accelerate the training algorithm, the proposed cost function to be minimized is the Cross Entropy Cost Function [24] which is defined as follows:

$$C = \frac{1}{m}\sum_{k=1}^{m}[\zeta_k \ln o_k + (1-\zeta_k) ln(1-o_k)] \tag{11}$$

where m is the number of samples used in the optimization step, ζ_k is the label for each sample and o_k is the network output which corresponds to the predicted label as it is shown in Fig. 2. The possible labels are: 0 for non-leak observations and 1 for leak observations.

Stochastic Gradient Descent [33] is the optimization algorithm selected for adjusting the parameters. The error between the output neuron value (before the comparison with the threshold) and the sample label is backpropagated through the network for each leak observation ($m = 1$), and the weights are updated with a fixed learning rate. Therefore, the parameters to be adjusted in the network are the following:

- Number of hidden layers L
- Number of nodes per hidden layer I
- Weights for each node (bias absorption was implemented) $w_i{}^l$
- Learning rate η
- classification threshold μ.

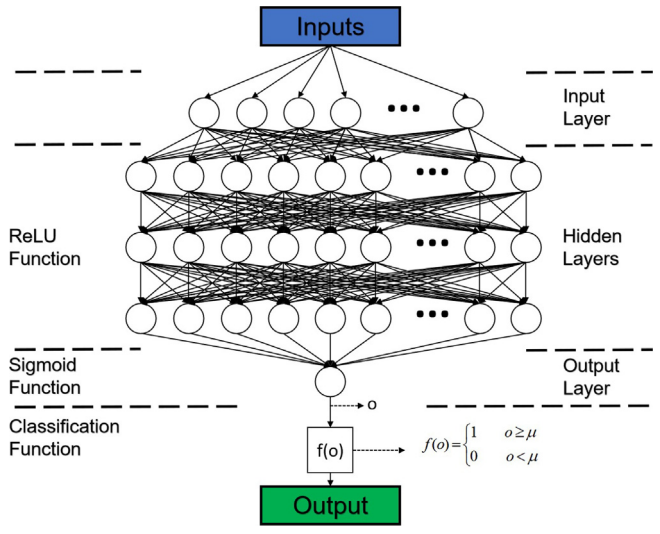


Fig. 2. Proposed neural network for leak detection.

In order to select the best architecture, 10-fold cross-validation was performed over a training set. The network architecture (depth and width) are selected through the following steps based on the performance measures selected for the problem:

1. A single hidden layer network is implemented.
2. The number of nodes per hidden layer is set equal to the number of input variables.
3. The best learning rate and classification threshold are adjusted by using grid search [42] (This procedure is followed for estimating the learning rate and the classification threshold for every architecture tested). Overfitting was prevented by the use of early stopping criterium [4].
4. The number of nodes in the hidden layer is increased by 5.
5. Steps 3 and 4 are repeated until the improvement in the performance measures selected for the leak detection task is not significant. This means that the difference in performance between one configuration and the other is not larger than a user-defined threshold.
6. The number of hidden layers is increased by one, and the number of hidden neurons is reset to the number of input variables for each layer.
7. Grid search is performed in order to determine the number of nodes in each hidden layer. The search space considered for each hidden layer in the grid search is I_{min} = number of inputs and
   I_{max} = best number of neurons found for a single hidden layer, with an increase of 5.
8. Steps 6 and 7 are repeated until the improvement in the performance measures selected for the leak detection task is not significant. Again, This means that the difference in performance between one configuration and the other is not larger than a user-defined threshold.

3.2. Leak size estimation

The leak size estimation task with data-driven methods consists of finding the value and confidence interval for the emitter coefficient C_e associated with the leak. Both tasks can be formulated as follows.

Definition 3.2 (*Data-driven estimation of leak size*). Given a data set $D_l = \{\mathbf{x}, C_e\}_{i=1}^{lc}$ ($\mathbf{x} \in \Re^m$) that is representative of the leaks whose size will be estimated, find a function $f_e(\mathbf{x}) : \Re^m \rightarrow C_e$ that maps any observation of the feature space to an emitter coefficient C_e associated with each leak.

Definition 3.3 (*Data-driven estimation of leak size range*). Given a data set $D_l = \{\mathbf{x}, C_e\}_{i=1}^{lc}$ ($\mathbf{x} \in \Re^m$) that is representative of the leaks whose size will be estimated, find a function $f_e(\mathbf{x}) : \Re^m \rightarrow C_{e_{int}}$ where $C_{e_{int}} = [C_{emin}, C_{emax}]$ that maps any observation of the feature space to a minimum and maximum possible values (C_{emin} and C_{emax}) of the emitter coefficient associated with the leak.

The feature vector $\mathbf{x}$ is formed by the residuals $\mathbf{r}$.

3.2.1. Gaussian processes

In this paper, a Gaussian Process Model is build to estimate, with a confidence interval, the values of the emitter coefficient of the detected leaks. A Gaussian process can be regarded as a probability distribution over functions that assumes every finite sample

of function values is jointly Gaussian distributed [31]. In particular, Gaussian process regression is used to estimate a function $g_*(.) = f_e(\mathbf{x})$ that best describes the data set D_l (with $X = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_{nc}}\}$ and $\mathbf{y} = \{C_e\}_{i=1} = \{y_1, y_2, \ldots, y_{nc}\}$) such that

$$y_i = g(\mathbf{x_i}) + \epsilon \quad \forall i \tag{12}$$

where $y_i \in \Re$ represents the emitter coefficient, and $\epsilon \sim N(0, \sigma_n^2)$ is a normally distributed Gaussian noise with mean zero and variance σ_n^2.

The Gaussian Process is completely characterized by a mean function $m(\mathbf{x})$ defined as the expected output value and a positive semi-definite covariance function $k(\mathbf{x}, \mathbf{x}')$ which defines the covariance between pairs of inputs (also called *Kernel* function) [14]. The output of the model can be described as follows:

$$g(\mathbf{x_i}) = \mathcal{N}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{13}$$
$$m(\mathbf{x}) \equiv \mathbb{E}\{g(\mathbf{x})\} \tag{14}$$
$$k(\mathbf{x}, \mathbf{x}') \equiv COV\{g(\mathbf{x}), g(\mathbf{x}')\} \tag{15}$$

In order to fit a Gaussian process model the following parameters must be adjusted:

- Kernel function $k(\mathbf{x}, \mathbf{x}')$
- Basis function $m(\mathbf{x})$
- Kernel scale σ_l
- Signal standard deviation σ_f

The Kernel scale σ_l and the signal standard deviation σ_f are adjusted for each combination of Basis-Kernel function through a grid search for maximizing the *log marginal likelihood* (10-fold cross-validation is used in this case) [31]. Once the model has been calibrated by adjusting the parameters of the covariance and mean functions, the inference of new output values from a set of test inputs is possible through the estimated function $g_*(.) \sim g(.)$. The output of this estimated function follows a multivariate Gaussian distribution and the probability distribution of the estimated latent function g_* is known. Therefore, a confidence region can be defined for any given output [31]. The resulting confidence interval for the emitter coefficient value is then used to ensure a smaller search space for the localization method.

3.3. Leak location

The leak localization in WDN, formulated like an inverse problem, was first proposed by Pudar in 1992 [28]. Recently, Steffelbauer et al. [39,40] also presented the leak localization as the solution of an inverse problem that is formulated as an optimization problem and solved by using the meta-heuristic algorithm Differential Evolution [41]. The main conclusion of their research is that the performance in the leakage localization task depends on the distance metric used to compute the objective function, as well as the labeling of the leakage positions in the parameter space (node of the network). In this paper, the leak location task (developed in a reduced space) is formulated as an inverse problem which is solved as an optimization problem by using a modified version of the Differential Evolution algorithm named Topological Differential Evolution with Temporal Analysis (TDE-TA). With this proposal, the location strategy does not depend on the node's labeling. This constitutes one of the contributions of this paper.

3.3.1. Formulation of the inverse problem

Let $\mathbf{m} \in \mathbb{R}^N$ be the vector of N flow and pressure measurements obtained from the sensors installed in the WDN and $\hat{\mathbf{m}} \in \mathbb{R}^N$ be the set of respective flow and pressure variables calculated from a WDN model. The vector $\hat{\mathbf{m}}$ is obtained by solving the system of Eqs. (1)–(4) for the following input variables.

- Nominal demand at the consumption nodes $\mathbf{d}$.
- Single leak in node p with emitter coefficient C_e. Both parameters will be included in the vector $\mathbf{s} = [p, C_e]$.

The solution of the system of equations mentioned above is represented here by $\hat{\mathbf{m}} = H(\mathbf{d}, \mathbf{s})$.

Finding the leak size and location through the inverse problem approach consists on determining the values for the vector $\mathbf{s}$ and $\mathbf{d}$ that minimize the difference between the measurements vector and the respective variables. In this paper, it is assumed that the demand vector is given such that only the vector $\mathbf{s}$ must be determined. This is formalized as follows.

Let Eq. (16) be a function that defines the similarity between the measurements vector ($\mathbf{m}$), and the estimated variables ($\hat{\mathbf{m}}$)

$$f(\mathbf{m}, \hat{\mathbf{m}}) = d(\mathbf{m}, \hat{\mathbf{m}}) \tag{16}$$

where $f(\mathbf{m}, \hat{\mathbf{m}})$ is called fitness or objective function and $d()$ is a similarity metric, the optimization problem to be solved is then defined as follows

$$\min_{\mathbf{s}} f(\mathbf{m}, H(\mathbf{d}, \mathbf{s})) \quad s.t. \quad C_{e_{min}} \leqslant C_e \leqslant C_{e_{max}} \quad 0 < p \leqslant P \tag{17}$$

where $C_e \in \mathbb{R}$ is the emitter coefficient, $C_{e_{min}}$ and $C_{e_{max}}$ are the possible minimum and maximum values that the emitter coefficient may take corresponding to the leak outflow to localize, $p \in \mathbb{N}$ is the leak position (the node where the leak occurs), and P is the number of nodes of the network where leaks might occur. The similarity metric considered in this paper is the Euclidean distance. Therefore, $d(\mathbf{m}, \hat{\mathbf{m}}) = \left(\sum_{i=1}^{N} (|m_i - \hat{m}_i|^2)\right)^{1/2}$.

This optimization problem can be solved by using meta-heuristics optimization methods which have demonstrated satisfactory performances by avoiding local minimums. The main reason to use this type of methods is that it cannot be demonstrated that the cost function is convex. Specifically, a modified version of the Differential Evolution optimization algorithm named Topological Differential Evolution with Temporal Analysis (TDE-TA) is proposed in this paper to find the solution.

3.3.2. Topological differential evolution with temporal analysis

Differential Evolution (DE) is an evolutionary algorithm based on populations. It allows to solve optimization problems with a friendly implementation structure. The process for finding a solution consists of three operators repeated until convergence: mutation, crossing and selection [3,41].

The solution model is usually expressed with the notation $DE/x/y/z$ (x: type of the mutation vector (also called the base vector), y: number of pairs of solutions to be used in order to vary the current solution, and z: distribution function which will be used during the crossing operation). The configuration $DE/rand/1/bin$ is considered in this paper where a random vector ($x = rand$) and the difference of a vector pair ($y = 1$) were applied in order to generate a mutation. A binomial crossing model ($z = bin$) is used for crossing two solutions. The general idea behind DE is to provide a new solutions by varying the properties of the members of the population. Each operator is mathematically described as follows.

- Mutation: Given a member of the population denoted by $\mathbf{s}^{(j,it)} = [C_e, p]$ at iteration it the mutation step is defined depending on the type of variable as follows
  1. For continuous parameters (C_e) the mutation step is

$$\bar{C}_e^{(j,it+1)} = C_e^{(\alpha,it)} + F_S(C_e^{(\beta,it)} - C_e^{(\gamma,it)}); \tag{18}$$

where $C_e^{(\alpha,it)}, C_e^{(\beta,it)}, C_e^{(\gamma,it)} \in \Re$ are members of the population generated at iteration it, and α, β and γ are randomly chosen in the interval $[0, N]$ with N being the population size. F_S is a real constant parameter known as scaling factor which determines the influence of the vector pair employed in the mutation.

2. For discrete parameters (p) the mutation step is based on the random selection of a node from the set formed by the current node and the respective neighboring nodes (connected to it through a pipe). This is mathematically defined by

$$\bar{p}^{(j,it+1)} = randi(\Theta^{(j,it)}) \tag{19}$$

where *randi* represents the random selection operator from a set of integer values, and $\Theta^{(j,it)} = \{p^{(j,it)}, p_1^{(j,it)}, p_2^{(j,it)}, \ldots, p_L^{(j,it)}\}$ such that $p_l^{(j,it)}$ is neighbor of $p^{(j,it)}$ and L is the total number of neighbors.

- Crossing: Once the vector $\bar{\mathbf{s}}^{(j,it+1)} = [\bar{C}_e^{(j,it+1)}, \bar{p}^{(j,it+1)}]$ is formed, crossing is considered according to

$$\bar{\bar{\mathbf{s}}}^{(j,it+1)} = \begin{cases} \bar{\mathbf{s}}^{(j,it+1)} & if \quad q_{rand} < C_R \\ \mathbf{s}^{(j,it)} & otherwise \end{cases} \tag{20}$$

where $0 < C_R < 1$ is the crossing constant, and q_{rand} is a random number sampled from a probability distribution function (binomial distribution in this paper).

- Selection:

$$\mathbf{s}^{(j,t+1)} = \begin{cases} \bar{\bar{\mathbf{s}}}^{(j,it+1)} & if \quad f(\mathbf{m}, H(\mathbf{d}, \bar{\bar{\mathbf{s}}}^{(j,it+1)})) < f(\mathbf{m}, H(\mathbf{d}, \mathbf{s}^{(j,it)})) \\ \mathbf{s}^{(j,it)} & otherwise \end{cases} \tag{21}$$

where $f(\mathbf{m}, H(\mathbf{d}, \bar{\bar{\mathbf{s}}}^{(j,it+1)}))$ and $f(\mathbf{m}, H(\mathbf{d}, \mathbf{s}^{(j,it)}))$ are the fitness function values for the newly generated vector $\bar{\bar{\mathbf{s}}}^{(j,it+1)}$ and the current vector $\mathbf{s}^{(j,it)}$, respectively.

Overall, the search vector for the leak location problem is formed by a real value (C_e) and a discrete value (p). If operators traditionally applied to continuous variables are applied to discrete variables then the node labeling influences the performance of the optimization method [39]. Therefore, the mutation step described above for discrete variables considers the topological properties of the WDN to sort this problem. This modification has been called topological mutation thus denoting the algorithm Topological-DE [34].

3.3.3. Temporal analysis and candidate set selection

It has been demonstrated in previous works that the leak location performance in WDN can be improved by performing a temporal analysis of the individual results [29,38]. Usually, the Bayes rule is applied over a window of samples based on the leak location probability estimated with machine learning tools. However, the application of this strategy when using the inverse problem approach is not straightforward. Therefore, a different strategy is proposed in this paper.

The evaluation of the cost function for each scenario provides valuable information that should be considered if the optimization problem is solved multiple times. One possible way to account for this information is to gather the final population of solutions obtained from each leak observation together with the values of the cost function achieved. Then, the minimum of all the solutions obtained is identified as the leak location solution.

Another way to exploit the search performed by the metaheuristic method is to analyze the set of solutions obtained. In this sense, instead of considering only one solution, a set of *nc* leak candidate solutions can be considered as possible locations. This set is determined by selecting the *nc* solutions corresponding to minimum values of the cost function among all populations obtained for different observations. While it may seem that this approach increases the uncertainty for the process operator regarding the leak location, it is likely that its adoption will actually reduce the leak location uncertainty because for large-scale networks with few sensors installed finding the exact location of the leak is very difficult in most cases. In fact, most authors evaluate the average distance between the true location and the one estimated. Therefore, assuming for example that the error threshold is two nodes in highly meshed networks the possible locations of the leak may increase significantly. Therefore, the approach adopted in this paper is to determine the number of candidate solutions that guarantee the exact location of the leak in the set of solutions obtained.

This approach is called Topological Differential Evolution with Temporal Analysis (TDE-TA) and the pseudocode of the complete algorithm is presented in Algorithm 1.

Algorithm 1. Pseudo-code for TDE-TA algorithm.

{**Inputs**: *Fs*: scaling factor, C_R: crossover constant, N: population size, T: time instants to analyze, **d**: demands of the network, T: number of time instants to analyze, ϱ: convergence threshold 1, ς: convergence threshold 2, *maxit*: maximum number of iterations, n_{lc}: number of leak node candidates to consider}
$S_T = \varnothing$ {Set of populations of all leak observations}
for all $t \in T$ **do**
{Initialization: generate each member of the population $S_t = \{\mathbf{s}^1, \ldots, \mathbf{s}^N\}$by sampling from a uniform random distribution by considering the respective intervals $0 < C_e < C_{e_{max}}$ (continuous) and $0 < p < P$ (discrete)} $it = 1$
repeat
for all $\mathbf{s}^j \in S_t$ **do**
{Mutation}
{For continuous variables according to 18}
{For discrete variables according to 19}
{Crossover according to 20}
{Selection according to 21}
end for
$it = it + 1$ **until** $|f(\mathbf{m}, H(\mathbf{d}, \mathbf{s}^{(j,it)})) - f(\mathbf{m}, H(\mathbf{d}, \mathbf{s}^{(j,it+1)}))| < \varsigma$ **or** $f(\mathbf{m}, H(\mathbf{d}, \mathbf{s}^{(j,it+1)})) < \varrho$ **or** $it == maxit$
$S_T = S_T \cup S_t$
end for
{Obtain the set *Ls* of n_{lc} leak candidate nodes with minimum cost function value from S_T (Omitting repeated entries)}
{**Output**: L_s set of leak candidate nodes}

4. Case study: Modena water distribution network

The case study considered in this paper is the model of the real water distribution network of the Italian city Modena shown in Fig. 3. This large-scale network is formed by 268 junctions connected through 317 pipes and served by 4 reservoirs [44].

4.1. Sensors: type, number and placement

The performance in the leak detection and location tasks is strongly dependent on the number of sensors installed in the network, their type and their position. In the case of WDNs, flow-rate, pressure head sensors or both can be considered. Researchers have devoted lots of efforts to the sensor placement problem to improve leak detection [12],30,47 and localization [5,8]. Most works con-

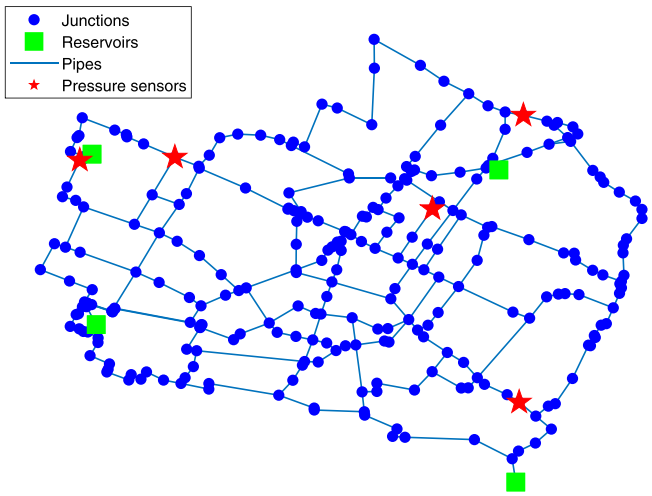


Fig. 3. Schematic representation of the Modena WDN.

sider only pressure head sensors because this type of sensors are cheaper and easier to install and maintain than flow meters. Moreover, in most real networks, the flow is measured at the output of each reservoir for operation management purposes. Therefore, both types of sensors are considered in this paper.

The number and accuracy of the sensors to be installed depends on the budget of the water company and the minimal leak size to be detected. Four flow sensors installed in the output of each reservoir are considered because this is a requirement for operation as explained before. Then, the number of pressure head sensors and location must be specified. The criteria considered to solve this task in this paper is the minimal size to be detected under nominal conditions (without considering the effect of any uncertainty in the parameters or consumer's demand).

The WaterGEMS tool was used for this task [2]. The number of leakage scenarios considered is 1000 (a rule of thumb is to select at least three times the number of network) and the leak size to be detected is 0.01 lps (liters per second). The sensor sensitivity is set to 0.01 m. Under these conditions, 97% of leakage coverage is reached and the pressure sensor's location is shown in Fig. 3 (Pressure sensors in nodes: 8, 107, 183, 240 and 253).

4.2. Simulation conditions

Matlab 2018a© and Epanet integrated through the EPANET-MATLAB-Toolkit were used for the simulation of the DMA [9]. Since it is unlikely that leaks with varying sizes will occur in all nodes of a network, a representative data set of real leaks is rarely available. Thus, the model of the network is generally used for generating synthetic leakage scenarios. If the simulation does not consider the operating condition of the real network and the leak data set ignores all possible locations the leak diagnosis methods will not have a satisfactory performance in real-life situations. It is therefore crucial to consider the stochastic conditions of the real system in the simulations. The main features accounted for the simulations of this paper are.

- Minimum flow regime (MFR) is considered. This consideration is made because in practice, it is very difficult to predict how the demand varies over the day. Conversely, it is relatively easy to characterize the demand during the early morning (2–6 am). Moreover, analysis under MFR has proven to be successful in real conditions.
- Leakages of different sizes are generated. The performance of the proposed approach is tested with leaks of small (Emitter coefficient – $\mathcal{N} \sim \{0.2, 0.1\}$), medium (Emitter coefficient – $\mathcal{N} \sim \{0.5, 0.25\}$), and large size – $\mathcal{N} \sim \{1, 0.25\}$). The leakage outflow varies between 1 lps and 2.5 lps for small and large leakages respectively. The total consumption of the network is 406.94 lps. Therefore, leaks generated represent between 0.2% and 0.6% of the total consumption.
- Stochastic demands are considered in the consumer's nodes to account for the typical uncertainty. Under MFR, it can be assumed that the demand is stationary with a Gaussian distribution. In this paper, the demand of each node is simulated according to the respective nominal consumption such that (Demand of node – $\mathcal{N} \sim \{d_n, 0.05 * d_n\}$ with d_n being the average consumption of the node n under MNF). Note that the uncertainty about the variability of each node depends on its nominal consumption.
- Measurement noise is added to the output signals according to the typical accuracy of the transducers used in the field (zero mean and 0.025 mH_2O of variance).
- Sampling time of 15 min is considered and hourly average values of the measurements are used aiming to reduce the effect of uncertainties.
- A total of 2000 scenarios are generated for the evaluation of the leak detection task, another 2000 for the leak size estimation, and finally, another 2000 for the leak location task. Each scenario contains 4 observations (4 h) generated under MFR and each observation is the average from 4 samples acquired from the network (sampling time is 15 min).

5. Results and discussion

This section presents the results of the different methods used in this paper for the leakage scenarios considered. Matlab 2018a© has been used for the implementation of the leak detection, leak size estimation, and leak location tasks.

5.1. Leak detection

The leak detection task can be framed as a binary classification problem as it was explained in Section 3.1. Thus, the performance measures to evaluate this task are generated from a confusion matrix represented in Table 1.

P and N represent the true total number of scenarios corresponding to leaks and no leaks, respectively. P' and N' represent the total number of scenarios identified by the detection algorithm as leaks and no leaks, respectively. TP (True positives) refers to the number of leak scenarios detected where a leak is truly occurring, TN (True negatives) refers to the number of no-leak scenarios identified where a leak is truly not occurring. FP (False positives) is the number of scenarios where a leak is detected but there is no leak occurring in the DMA, and FN (False negatives) is the number of scenarios where a leak is not detected. From the confusion matrix the following performance measures can be derived:

- Sensitivity or Leak detection rate: $LDR = TP/P$.
- Specificity: $Spec = TN/N$.
- False alarm rate: $FAR = FP/N$.
- Accuracy: $Acc = (TP + TN)/(P + N)$.

Other useful performance indicator is area under the curve (AUC) of the receiver operating characteristic plot or ROC curve. Such plot captures the trade-off between leak detection and false alarm avoidance. The larger the AUC is, the better is the performance of the monitoring model. Note that the latency or leak detection delay is not analyzed because of two reasons: the long sampling time considered to analyze one observation (1 h) and the short period of time during which the leak detection task is

Table 1
Confusion matrix.

		Predicted class		
		Leak	No leak	Total
Actual class	*Leak*	TP	FN	P
	No leak	FP	TN	N
	Total	P'	N'	P + N

performed (4 observations). In the case of the performance measure FAR, it is necessary to highlight that lower values of the index indicate better results. The performance results of all methods were compared through the Wilcoxon signed rank test [20].

Two data sets are generated one for training and the other for testing. Each data set contains data of nominal conditions (class 0) and data of small and large leaks (class 1). A total of 1000 observations are included for each data set where the node location is randomly generated and the nominal conditions of the network were simulated as described previously but no leak is present. The training of the neural networks is based on 10-fold cross validation according to the procedure described in Section 3.1 and the architecture and parameters obtained are:

- Number of hidden layers $L = 3$
- Number of neurons in each layer $I = 20$
- Learning rate $\eta = 0.006$
- Detection threshold $\mu = 0.09$

The leak detection task with the proposed neural network will be compared against four tools which can be used to solve binary classification problems: Principal Component Analysis (PCA) [26], a univariate statistical test [17], Linear Discriminant Analysis (LDA), and Linear Support Vector Machines (L-SVM).

Univariate statistical tests are based on the characterization of the statistical distribution properties of the variables that represent the nominal behavior of the system. For instance, in [17] outlier regions are defined to detect anomalous behaviors. Since this paper only considers minimum night flow conditions, a normal distribution can be assumed such that a t-test to assess the equality of the mean is used. Nonetheless, one weakness of the univariate analysis is that it does not considers the relationship among variables to make a decision.

PCA has been previously used for leak detection with successful results [26]. However, PCA allows to model the linear relations among the variables which are implicit in the covariance matrix of the data representing the nominal behavior of the system. Similarly, LDA allows to model the linear relationships among the variables but unlike PCA, it requires the information about classes to be known a priori. Moreover, while PCA captures the information of the variables associated with their variability, LDA explicitly tries to capture the difference between classes based on the variables' discriminant power. Finally, L-SVM tries to find the best maximum-margin hyperplane which separates the two classes by solving an optimization problem.

As explained above, since there are non-linear relations for the DMA variables all the classification methods mentioned above may not provide the best possible results. It must be highlighted then that other binary classification tools for non-linear data could be used for the leak detection task, i.e. Support Vector Machines. Nonetheless, the goal of the proposed approach is just to accomplish a satisfactory performance in the leak detection task.

The testing data set is divided into 10 parts and the mean and standard deviation of the performance measure are calculated. The results of the leak detection task for the testing data sets is presented in Tables 2–4. Only one testing nominal data set has been considered for the evaluation of the specificity and false alarm rate performance measures. Therefore, the results are divided into three tables, one for nominal operating conditions, one for small leaks, and one for medium-large leaks. The Accuracy and AUC for Tables 3 and 4 are calculated based on the combination of nominal data set with small leak data set and large leak data set, respectively. The mean and standard deviation of each indicator is expressed as "mean (std)".

First,the FAR (Specificity is equivalent) of the different methods is analyzed. The reliability of overall leak monitoring strategy mainly depends on the FAR because many false alarms will harm the confidence of the operators in the application. Therefore, the results are quite satisfactory in this sense for both the PCA, LDA, and DNN leak detection methods. For the univariate tests, the FAR reaches more than 5%, which is considered as unacceptable for most applications. This also occurs for L-SVM. In the latter case, the optimization process to achieve the best possible classification performance does not take into account which class should be prioritized to be correctly classified.

The LDR performance for medium to large leaks is always satisfactory for all methods. However, the detection of small leaks is difficult because the variability of the demand in some nodes is high due to their nominal consumption variability. In this sense, the DNN outperforms all other methods. The poor performance of the LDA method owes to the violation of one the assumptions of this classifier (each variable has the same variance). The AUC and Acc can be considered as the most complete performance measures in the sense that they consider the balance between false alarms and leak detection rate. As it can be observed in the tables, the proposed DNN clearly outperforms all other methods if the AUC and Acc indices are considered. Other non-linear binary classification methods such as Support Vector Machines may be able to cope with this task achieving a similar performance than the DNN.

5.2. Leak size estimation

Only the leaks which are detectable are considered for developing the Gaussian process estimator. The variable to be estimated is the emitter coefficient C_e that corresponds to the detected leakage. Note that there is a non-linear relationship between the pressures and flows of the network and the emitter coefficient according to Eq. (4). The main goal of the estimator is to calculate the leak size $\hat{C_e}$ together with the leak size interval: minimum $\hat{C_{e_{min}}}$ and maximum $\hat{C_{e_{max}}}$. In other words, the possible values that the leak may take. Therefore, the leak size estimation is framed as a regression problem and the performance measures considered for the leak estimation method are.

- Root mean squared error: $RMSE = \sqrt{\frac{\sum_{i=1}^{N}(C_e - \hat{C_e})^2}{N}}$.
- Mean Absolute Error: $MAE = \frac{\sum_{i=1}^{N}|C_e - \hat{C_e}|}{N}$.
- Hits: Number of scenarios where the true leak size is within the leak size interval estimated divided by the total number of scenarios.

Table 2
Leak detection performance for nominal operation conditions (Indicators are expressed in percent as "mean (std)").

	Leak detection methods				
Index	Univariate test	PCA	LDA	L-SVM	DNN
Spec	94.4 (1.26)	**99.6 (0.7)**	**100**	91.04 (1.63)	**99.04 (0.56)**
FAR	5.6 (1.26)	**0.4 (0.7)**	**0**	8.96 (1.63)	**0.96 (0.56)**

Table 3
Leak detection performance for medium-large leakages (Indicators are expressed in percent except AUC).

	Leak detection methods				
Index	Univariate test	PCA	LDA	L-SVM	DNN
LDR	**100 (0)**	**100 (0)**	**99.24 (0.4)**	**100 (0)**	**99.8 (0.19)**
Acc	**99.07 (0.21)**	**99.93 (0.15)**	**99.62 (0.2)**	95.89 (0.5)	**99.55 (0.21)**
AUC	0.972 (0.0063)	**0.998 (0.0035)**	**0.996 (0.002)**	0.96 (0)	**0.999** (10^{-5})

Table 4
Leak detection performance for small leakages (Indicators are expressed in percent except AUC).

	Leak detection methods				
Index	Univariate test	PCA	LDA	L-SVM	DNN
LDR	88.12 (0.23)	84.74 (1.08)	65.12 (0.84)	**92.2 (0.69)**	88.9 (2.26)
Acc	89.17 (0.28)	87.22 (0.8)	82.56 (0.42)	91.62 (0.61)	**93.97 (1.12)**
AUC	0.913 (0.0064)	0.923 (0.003)	0.83 (0)	0.92 (0.0061)	**0.976 (0.0032)**

The first and second indicators give an idea of how accurate the estimator is. However, the third indicator is the most important one for this application because the leak location method will try to find the leak location based on this range according to the optimization problem formulated in (17). If the estimated range is wrong, the leak location method will likely identify the wrong location for the leakage.

A new data set of leaks is generated for training and 10 data sets were generated for testing. Both small and medium-large leaks are included for a total of 1000 leak scenarios in each data set. Before using each one, it is pre-processed with the leak detection method to determine which leaks are detectable.

The following three Kernel functions were considered:

- Exponential Kernel Function

$$k(x,x') = \sigma_f^2 exp\left(-\frac{r}{\sigma_l}\right)$$

- Squared Exponential Kernel Function

$$k(x,x') = \sigma_f^2 exp\left(-\frac{1}{2}\frac{r^2}{\sigma_l^2}\right)$$

- Matern Kernel Function with parameter 5/2

$$k(x,x') = \sigma_f^2\left(1+\frac{\sqrt{5}r}{\sigma_l}+\frac{5r^2}{3\sigma_l^2}\right)exp\left(-\frac{\sqrt{5}r}{\sigma_l}\right)$$

where r is the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}'$.

The following two basis functions were considered:

- Constant Basis Function
- Linear Basis Function.

The parameters were adjusted for each possible combination through 10-fold cross-validation by using Matlab R2018a *Hyperparameter Optimization* feature in the *Gaussian process regression* fitting tool. A 95% confidence interval is considered for the estimation of the leak size interval. The performance measure considered to select the best combination of parameters is *hit* rate. The constant basis function yielded better performance than the linear basis function for all scenarios. A comparison on the three Kernel functions when using constant basis function is shown in Table 5. The mean and standard deviation of each indicator is expressed as "mean (std)".

A statistical comparison of the three Kernels based on the 10-fold cross validation results by considering the Hits performance measure. The Wilcoxon signed rank test was used to determine if significant differences were present. It was obtained with $\alpha = 0.05$ that there are no significant differences between the Exponential and Squared Exponential Kernels both outperforming the Mattern Kernel. This indicates that any of those Kernels could be used for the leak size estimation task.

The leak size estimation problem can also be framed as a multi-class classification problem. This can be achieved if the data is divided into classes according to the leak size interval (the smaller the interval selected the larger the number of classes created). An interval similar to the average estimated emitter coefficient interval size obtained through Gaussian Process Regression is selected for fair comparison purposes. Therefore, a shallow artificial neural network (SNN) and a support vector machine classifier with Gaussian kernel (GSVM) were trained to classify the samples into the

Table 5
Leak estimation results for different Kernels based on RMSE and Hits indexes ("mean (std)").

Kernel	Hits	RMSE	MAE
Exponential	94.47 (0.38)	0.05 (0.0004)	0.04 (0.0004)
Squared Exponential	96.79 (2.87)	0.08 (0.04)	0.06 (0.04)
Matern	93.94 (3.3)	0.06 (0.007)	0.04 (0.006)

five classes (each one corresponding to a different leak size interval).

The implementations of SNN and GSVM in Matlab 2018a© were used. Bias absorption was used for the SNN and the Levenberg–Marquardt algorithm was employed. Tan-sig activation function was used for hidden layers and soft-max for the output layer. The number of neurons in the single hidden layer was adjusted through 10-fold cross-validation and 20 neurons allowed to achieve the best performance. For the GSVM classifier, the Kernel Scale and Box Constraint hyper-parameters were optimized by using grid search. Both One vs. One and One vs. All were tested showing no statistical differences. The performance in terms of Hits for both methods is presented in Table 6.

Overall, the Hits index reflects a poor performance for both classifiers compared to the performance achieved by the GP. For the latter one, the leak size interval is correctly estimated in around 95% of the total scenarios. These results were verified through Wilcoxon signed rank test. Perhaps a more complex classifier may achieve a better performance but still, one of the contributions of the paper is the idea of using a tool to estimate a leak size interval. This is because the possibility of estimating the leak size interval accurately will contribute to the adequate leakage location, as it will be shown in the next subsection.

5.3. Leak location

The following parameters are used in the execution of the TDE-TA optimization algorithm:

- C_R= 0.9 for ensuring a low influence of the population solution obtained up to that moment
- F_S = 0.6, which decides the influence of the mutation operator.
- The population is formed by 20 individuals ($N = 20$).
- The stopping criteria established are convergence threshold for optimization function difference $\varrho = 10^{-4}$, convergence threshold for cost function value $\varsigma = 10^{-6}$, and maximum number of iterations $maxit = 100$.
- The metric used to formulate the optimization function is the Euclidean Distance as stated before.

In practice, most inference methods based on a limited number of sensors in large-scale networks, such as the case presented in this paper, allow to determine the approximate leak location. Once a candidate of possible locations is identified, the operator of the water management plant will use specialized equipment (i.e. acoustic loggers) to narrow down the exact location of the leak. Therefore, it is desirable to define the candidate set of nodes where the leak is present with highest accuracy because this will save time in the process of searching the exact location. For instance, if only one node is identified by the location method and the true location is one or two nodes away, the operator will spend more time identifying the true location. Conversely, if a candidate set of more nodes is considered, and the true location is most of the times within that set, the operator will generally spend less time searching for the leakage.

Table 6
Leak estimation results for a SNN and GSVM classifiers -Hits index ("mean (std)").

Method	Hits
SNN	75.86 (1.14)
GSVM	77.97 (1.00)

The performance measures considered to evaluate the leak location task are.

- Exact location of leaks:

$$EL = \left(\sum_{i=1}^{N_S} E_i\right)/N_S \quad 0 \leqslant EL \leqslant 1 \tag{22}$$

where N_S is the number of leak scenarios considered and

$$E_i = \begin{cases} 1 & if \quad p_i \in Ls_i \\ 0 & otherwise \end{cases} \tag{23}$$

where p_i represents the true node location and Ls_i is the set of node candidates estimated for scenario i.

- Location of leaks with a distance of one node to the true location:

$$ONL = \left(\sum_{i=1}^{N_S} ON_i\right)/N_S \quad 0 \leqslant ONL \leqslant 1 \tag{24}$$

$$ON_i = \begin{cases} 1 & if \quad d_{top}(p_i, p_j) = 1 \text{ with } p_j \in Ls_i \\ 0 & otherwise \end{cases} \tag{25}$$

where p_i represents again the true node location, p_j represents any node from the set of node candidates Ls_i, and d_{top} is the topological distance between two nodes in terms of number of links or pipes.

The leak location performance depends, among other things, on the size of set candidate nodes considered Ls_i. If the candidate set is reasonably increased such that it can be guaranteed that the true leak node will be among the candidates the reliability of the leak location method is improved. The performance in the leak location task also depends on the number of the observations considered in the temporal analysis.

Only the scenarios with leaks of medium-large size are considered to evaluate the leak location performance. The two main reasons are: 1) it is more important to locate medium to large leakages because the water losses are more significant, and 2) achieving a satisfactory location performance of small leakages with the number of sensors and case study considered in this work is not feasible. To locate small leakages in large-scale networks from 20 to 30 sensors are required. For instance, in [48,49], 20 to 40 sensors are required to achieve a good leak location performance for a WDN with 375 nodes. However, water companies usually cannot afford that many sensors.

To analyze the performance of the leak location approach a set of 2000 leak scenarios of medium-large leaks where generated (each one considering up to 4 observations). The data set is divided into 10 parts to evaluate the mean and standard deviation of the performance measures. The results for different sizes of the candidate set and number of observations analyzed (window size) are shown in Table 7 for the *EL* performance measure and in Table 8 for *ONL*. The mean and standard deviation of each indicator is expressed as "mean (std)".

As it can be observed from both tables, the performance obtained when considering only one node is not satisfactory for the exact node location criteria (*EL*). However, for the one-node distance criteria the performance reaches 0.77 when the window size is four observations (which is the maximum that can be considered for minimum night flow conditions). Nonetheless, when the number of candidate nodes also increases an improvement in both performance measures is observed. Therefore, the leak location performance improves both when more observations are analyzed and when more candidate nodes are considered. The best performance achieved for the exact node location criteria is

Table 7
Leak location performance for the exact node location indicator *EL* (“mean (std)”).

T	Number of candidates in *Ls*				
	1	2	3	4	5
1	0.40 (0.036)	0.56 (0.034)	0.64 (0.044)	0.66 (0.037)	0.67 (0.038)
2	0.49 (0.032)	0.66 (0.026)	0.73 (0.026)	0.78 (0.021)	0.80 (0.023)
3	0.50 (0.031)	0.69 (0.028)	0.78 (0.025)	0.82 (0.026)	0.86 (0.023)
4	0.52 (0.035)	0.70 (0.042)	0.79 (0.035)	0.84 (0.026)	**0.87 (0.018)**

Table 8
Leak location performance for a distance of one node to the true location *ONL* (“mean (std)”).

T	Number of candidates in *Ls*				
	1	2	3	4	5
1	0.63 (0.039)	0.71 (0.044)	0.75 (0.050)	0.76 (0.050)	0.77 (0.049)
2	0.72 (0.024)	0.80 (0.015)	0.84 (0.015)	0.87 (0.025)	0.88 (0.020)
3	0.75 (0.025)	0.84 (0.021)	0.89 (0.019)	0.91 (0.018)	0.93 (0.016)
4	0.77 (0.037)	0.86 (0.030)	0.90 (0.020)	0.92 (0.018)	**0.94 (0.019)**

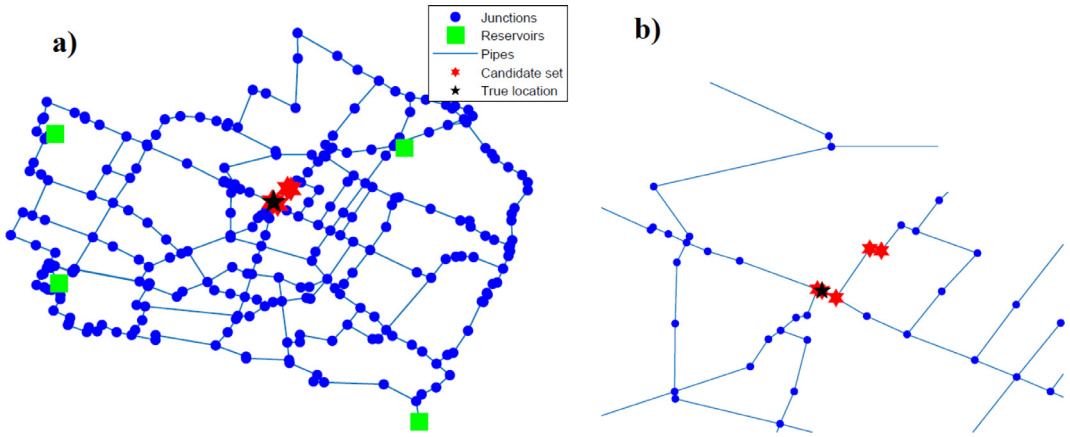


Fig. 4. Leak location scenario for a case where the true node is identified within the candidate set of 5 nodes: a) Network, b) Zoom of the specific area.

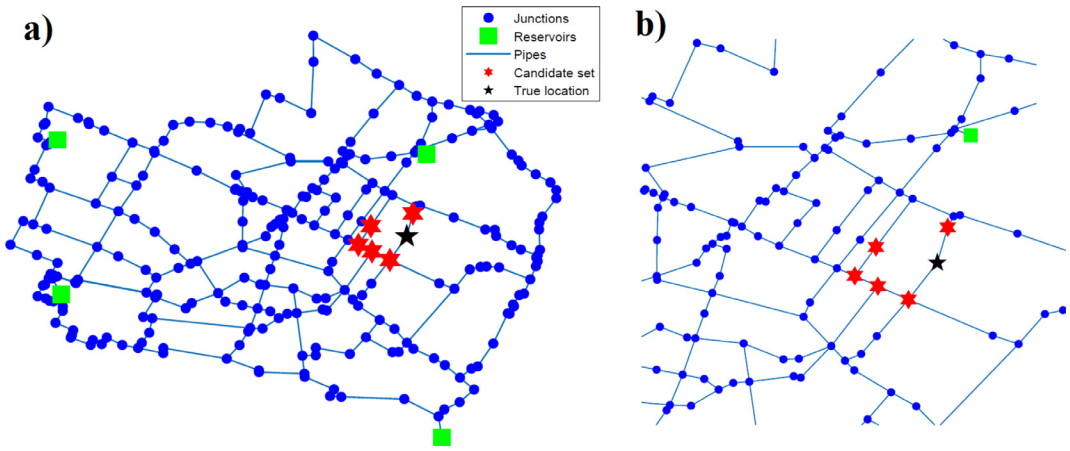


Fig. 5. Leak location scenario for a case where the true node is not within the candidate set of 5 nodes: a) network, b) Zoom of the specific area.

$EL = 0.87$ which can be interpreted such as for every 10 leaks 9 are will be located exactly within the candidate set estimated. If the candidate set is further increased with the nodes adjacent to the five nodes then the performance increases to 0.94 (*ONL* indicator).

The candidate set of nodes identified from the population of solutions obtained from TDE-TA are in general located close to one another. This means that leakage area can be narrowed down easily by the process operators. Figs. 4 and 5 show four candidate

Table 9
Comparison of leak location performance for the exact node location indicator *EL* with and without leak size search space reduction (window size of 4 observations).

Red.	Number of candidates in *Ls*				
	1	2	3	4	5
Yes	0.52 (0.035)	0.70 (0.042)	0.79 (0.035)	0.84 (0.026)	0.87 (0.018)
No	0.5 (0.079)	0.69 (0.094)	0.78 (0.061)	0.84 (0.046)	0.87 (0.023)

Table 10
Comparison of leak location performance for a distance of one node to the true location *ONL* with and without leak size search space reduction (window size of 4 observations).

Red.	Number of candidates in *Ls*				
	1	2	3	4	5
Yes	0.77 (0.037)	0.86 (0.030)	0.90 (0.020)	0.92 (0.018)	0.95 (0.019)
No	0.77 (0.079)	0.87 (0.094)	0.92 (0.061)	0.93 (0.046)	0.95 (0.023)

Table 11
Comparison of different leak location methods.

	Features to compare			
Method	Nodes	Sensors	Candidate nodes (Size)	Accuracy (%)
Zhang et al. 2016 [49]	375	10–25	15–75	90–97
Xie et al. 2019 [48]	375	20–40	9–19	77–97
Raei et al. 2019 [30]	1588	6–35	39–397	60–98
Proposal	368	9	1–5	52–87

sets for different leak scenarios. As it can be noted in both scenarios, the candidate set identified by the location algorithm is geographically close. This can be explained by the way the metaheuristic algorithm operates. Moreover, even in the case where the leak is not exactly located within the candidate set, it is located within one-node distance of the true leak node.

The reduction of the leak size search space to solve the inverse problem improves the leak location performance. This is illustrated in Tables 9 and 10. Moreover, the computational time consumed to estimate the leak range is not critical once the Gaussian Process function is obtained because the sampling times of the WDN are generally very large in the order of 1 h for the leak location analysis (compare to the time it takes to estimate the leak size in the order of seconds). As it can be appreciated in both tables the average performance is similar while the use of a reduced leak size interval also reduces the standard deviation of the results. Therefore, this contributes to a more accurate leak location.

Finally, we highlight that previous methods applied for leak detection and location were commonly evaluated on private commercial data sets or on private models of water distribution networks, and as a result, it is not possible to directly compare these methods with the proposed approach. Thus, since there is no straightforward way to make this comparison, it is considered to perform a comparison based on the following metrics:

- Number of nodes of the network. This metric is related to the complexity and size of the network.
- The number of sensors to install is determined by the budget of the water company. The more sensors to install the more expensive is the inversion because of the costs of buying the sensors, their installation, operation, as well as posterior maintenance.
- The number of candidate nodes. This metric determines the number of nodes that the method presents as a possible location for the leakage. If the candidate nodes is high, then there is more uncertainty regarding the true location of the leakage. This means that more effort is required to narrow down the exact location through specialized equipment. Therefore, it is desirable that the candidate nodes is as small as possible to allow the fast location of the leakages.
- The accuracy expresses the number of times that the method makes the wrong choice about the area (candidate nodes) where the leak is located.

Table 11 shows this comparison. As it can be appreciated, the proposed approach can reach a satisfactory performance (around 87% for accuracy) with small amount of sensors (9) and small set of candidate modes (up to 5). A smaller candidate set translates into a smaller leak area identified by the method. Moreover, the proposed approach requires less sensors than the other approaches. Therefore, overall, it is a more cost-efficient solution.

6. Conclusions

In this paper, a new approach for leak detection, estimation and localization that combines data-driven and model-based methods is presented. The proposal is tested with the model of a large-scale real WDN belonging to the Italian city Modena by considering only 4 flow meters installed in the reservoirs of the network and 5 pressure head sensors.

In the detection stage, a deep neural network is used achieving a detection performance of around 90% considering small, medium and large size leakages which represents a remarkable result considering the high complexity of the WDN and the small number of sensors used.

In the estimation stage, a Gaussian process regression is used to estimate the leakage size range with a performance of 0.05 (RMSE). Moreover, the true value of the leak size is within the range estimated in 95% of all scenarios considered. The benefit of using this estimation for the leak location is also demonstrated when this task is solved as an inverse problem. This result is achieved because the leak size range estimation allows to reduce the search space for the leak size.

Finally, the leak localization task is solved as an inverse problem. A modified version of the Differential Evolution optimization algorithm named Topological Differential Evolution with Temporal Analysis (TDE-TA) is used for this purpose. The TDE-TA algorithm considers the topological structure of the WDN to modify the solutions that form the population. Two important modifications incorporated to find the leak location are 1) to perform a temporal analysis of the solutions found for each scenario, and 2) to estimate a candidate set of solutions based on the final population estimated with TDE-TA. Both modifications are based on the value of the cost function of each solution such that the minimum of all members of the population are considered. Therefore, a set of candidate solutions is created based on the minimum values of cost function over time. Thanks to the topological evolution the resulting candidate set creates a zone of reduced possible locations which are geographically close to one another and this is very useful from a practical point of view.

Future research will be focused on finding new evolution mechanisms that allow to find the leak zone straightforwardly.

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.

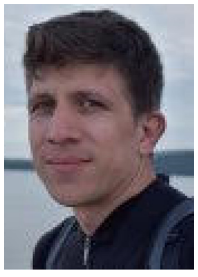
Acknowledgements

The authors would like to thank the anonymous reviewers because their thought-provoking and insightful comments and corrections have been very useful for improving the paper quality. The authors also acknowledge the decisive support provided by CAPES- Foundation for the Coordination and Improvement of Higher Level Education Personnel, through the project "Computational Modelling for Applications in Engineering and Environment", Program for Institutional Internationalization CAPES PrInt 41/2017, Processo No. 88887.311757/2018-00. Our gratitude is also due to the Brazilian Society of Computational and Applied Mathematics (SBMAC), to CNPq – National Council for Scientific and Technological Development, to FAPERJ – Foundation Carlos Chagas Filho for Research Support of the State of Rio de Janeiro, as well as to the Cuban Ministry of Higher Education (MES) and Universidad Tecnológica de La Habana José Antonio Echeverría, CUJAE.

References

[1] C. Arsene, B. Gabrys, D. Al-dabass, Decision support system for water distribution systems based on neural networks and graphs theory for leakage detection, Expert Syst. Appl. 39 (18) (2012) 13214–13224.
[2] Bentley Systems Incorporated, 2006. WaterGEMS v8 Users Manual..
[3] L. Camps Echevarría, O. Llanes-Santiago, A.J. da Silva Neto, An approach for fault diagnosis based on bio- inspired strategies, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010. No. September 2015, 2010, IEEE, pp. 1–7. https://ieeexplore.ieee.org/document/5586357..
[4] R. Caruana, S. Lawrence, L. Giles, Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping, in: Proceedings of the 13th International Conference on Neural Information Processing Systems, NIPS'00, MIT Press, Cambridge, MA, USA, 2000, pp. 381–387. URL http://dl.acm.org/citation.cfm?id=3008751.3008807.
[5] M. Casillas, V. Puig, L. Garza-Castañón, A. Rosich, Optimal sensor placement for leak location in water distribution networks using genetic algorithms, Sensors (Basel) 13 (11) (2013) 14984–15005.
[6] A.L. Caterini, D.E. Chang, Deep Neural Networks in a Mathematical Framework, Springer, 2018.
[7] D. Covas, H. Ramos, A.B. de Almeida, Standing wave difference method for leak detection in pipeline systems, J. Hydraul. Eng. 131 (12) (2005) 1106–1116.
[8] M.À. Cugueró-Escofet, V. Puig, J. Quevedo, Optimal pressure sensor placement and assessment for leak location using a relaxed isolation index: application to the Barcelona water network, Control Eng. Pract. 63 (2017) 1–12.
[9] D. Eliades, M. Kyriakou, S. Vrachimis, M. Polycarpou, EPANET-MATLAB toolkit: an open-source software for interfacing EPANET with MATLAB, in: Proceedings of the 14th International Conference on Computing and Control for the Water Industry, CCWI, 2016, The Netherlands, p. 8..
[10] E. Farah, I. Shahrour, Leakage detection using smart water system: combination of water balance and automated minimum night flow, Water Resour. Manage. 31 (15) (2017) 4821–4833.
[11] M. Farley, S. Trow, Losses in Water Distribution Networks A Practitioner's Guide to Assessment, Monitoring and Control, IWA Publishing, London, UK, 2003.
[12] M. Hagos, D. Jung, K.E. Lansey, Optimal meter placement for pipe burst detection in water distribution systems, J. Hydroinf. 18 (4) (2016) 741–756.
[13] R. Houghtalen, A. Akan, N. Hwang, Fundamentals of Hydraulic Engineering Systems, 4th Edition., Prentice Hall, Englewood Cliffs, NJ, 2010.
[14] M.F. Huber, Recursive Gaussian process: on-line regression and learning, Pattern Recogn. Lett. 45 (1) (2014) 85–91.
[15] D. Laucelli, M. Romano, D. Savic, O. Giustolisi, Detecting anomalies in water distribution networks using EPR modeling paradigm, J. Hydroinf. 18 (3) (2016) 409–427.
[16] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (19) (2017).
[17] 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. 13 (3) (2016) 242–252.
[18] M. Loveday, J. Dixon, DMA sustainability in developing countries, in: Proc. Leakage 2005 Conference, 2005, Halifax, Canada..
[19] Z. Lu, H. Pu, F. Wang, Z. Hu, L. Wang, The Expressive Power of Neural Networks: A View from the Width. Tech. rep., Cornell University, 2017. URL https://arxiv.org/abs/1709.02540.
[20] J. Luengo, S. García, F. Herrera, A study on the use of statistical tests for experimentation with neural networks: analysis of parametric test conditions and non-parametric tests, Expert Syst. Appl. 36 (2009) 7798–7808.
[21] B. Mergelas, G. Henrich, Leak locating method for precommissioned transmission pipelines: North American case studies, in: Proc. Leakage 2005 Conference, Halifax, Canada, 2005..
[22] S. Mounce, R. Mounce, J. Boxall, Novelty detection for time series data analysis in water distribution networks systems using support vector machines, J. Hydroinf. 13 (4) (2011) 47–54.
[23] J.M. Muggleton, M.J. Brennan, Leak noise propagation and attenuation in submerged plastic water pipes, J. Sound Vib. 278 (2004) 527–537.
[24] G. Nasr, E. Badr, C. Joun, @inproceedingsNasr2002CrossEE, title=Cross Entropy Error Function in Neural Networks: Forecasting Gasoline Demand, author=George E. Nasr and Elie A. Badr and C. Joun, booktitle=FLAIRS Conference, year=2002, in: FLAIRS Conference, 2002..
[25] C. Palau, F. Arregui, M. Carlos, Burst detection in water networks using principal component analysis, J. Water Resour. Plann. Manage. 138 (1) (2012) 47–54.
[26] C.V. Palau, F.J. Arregui, M. Carlos, Burst detection in water networks using principal component analysis, J. Water Resour. Plann. Manage. 138 (1) (2012) 47–54.
[27] R. Pérez, V. Puig, J. Pascual, J. Quevedo, E. Landeros, A. Peralta, Methodology for leakage isolation using pressure sensitivity analysis in water distribution networks, Control Eng. Pract. 10 (10) (2011) 1157–1167.
[28] R. Pudar, J. Liggett, Leaks in pipe networks, J. Hydraul. Eng. 118 (1992) 1031–1046.
[29] M. Quiñones-Grueiro, J.M. Bernal-de Lázaro, C. Verde, A. Prieto-Moreno, O. Llanes-Santiago, Comparison of classifiers for leak location in water distribution networks, IFAC-PapersOnLine 51 (24) (2018) 407–413.
[30] E. Raei, M.R. Nikoo, S. Pourshahabi, M. Sadegh, Optimal joint deployment of flow and pressure sensors for leak identification in water distribution networks, Urban Water J. 15 (9) (2019) 837–846.
[31] C.E. Rasmussen, Advanced lectures on machine learning, Lecture Notes in Computer Science 3176: Lecture Notes in Artificial Intelligence. Springer, Berlin, Heidelberg, Ch. Gaussian processes in machine learning, 2004, pp. 63–71..
[32] L.A. Rossman, Epanet 2 Users Manual, 2000..
[33] S. Ruder, An overview of gradient descent optimization algorithms. CoRR abs/1609.04747, 2016. http://arxiv.org/abs/1609.04747..
[34] M. Sánchez-Rivero, M. Quiñones Grueiro, C. Corona Cruz, A. Silva Neto, O. Llanes-Santiago, SOCO 2019: 14th International Conference on Soft Computing Models in Industrial and Environmental Applications. Vol. 950 of Advances in Intelligent Systems and Computing. Springer, Springer, Cham, Ch. A Proposal of Robust Leak Localization in Water Distribution Networks Using Differential Evolution, 2020, pp. 311–320..
[35] G. Sanz, R. Pérez, Z. Kapelan, D. Savic, Leak detection and localization through demand components calibration, J. Water Resour. Plann. Manage. 142 (2) (2016) 1097–1098.
[36] S. Skansi, Introduction to Deep Learning. Undergraduate Topics in Computer Science, Springer International Publishing, 2018.
[37] A. Soldevila, J. Blesa, S. Tornil-Sin, A. Duviella, R. Fernández-Canti, V. Puig, Leak location in water distribution networks using a mixed model-based/data-driven approach, Control Eng. Pract. 55 (2016) 162–173.
[38] A. Soldevila, R. Fernández-Canti, J. Blesa, S. Tornil-Sin, V. Puig, Leak localization in water distribution networks using bayesian classifiers, J. Process Control 55 (2017) 1–9.

[39] D.B. Steffelbauer, M. Günther, D. Fuchs-Hanusch, Leakage localization with differential evolution: a closer look on distance metrics, Procedia Eng. 186 (2017) 444–451.
[40] D.B. Steffelbauer, M. Günther, M. Neumayer, D. Fuchs-Hanusch, Sensor placement and leakage isolation with differential evolution, in: World Environmental and Water Resources Congress. Portland, U.S., 2014, pp. 408–416.
[41] R. Storn, K. Price, Differential evolution a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, Int CS Institute, University of California, 1995..
[42] I. Syarif, A. Prugel-Bennett, G. Wills, Svm parameter optimization using grid search and genetic algorithm to improve classification performance, TELKOMNIKA 14 (4) (2016) 1502–1509.
[43] UN, 2008. World Urbanization Prospects: The 2007 Revision Population Database. URL http://esa.un.org/unup/.
[44] Q. Wang, M. Guidolin, D. Savic, Z. Kapelan, Two-objective design of benchmark problems of a water distribution system via MOEAs: towards the best-known approximation of the true Pareto front, J. Water Resour. Plann. Manage. 141 (3) (2015) 1–14.
[45] Y. Wu, S. Liu, A review of data-driven approaches for burst detection in water distribution systems, Urban Water J. 14 (9) (2017) 1–12.
[46] Y. Wu, S. Liu, X. Wu, L. Y., Y. Guan, Burst detection in district metering areas using a data driven clustering algorithm, Water Res. 100 (2016) 28–37..
[47] Z.Y. Wu, Y. Song, E. Roshani, Software prototype for optimization of monitoring and data logging in water distribution systems, Procedia Eng. 119 (1) (2015) 470–478.
[48] X. Xie, D. Hou, X. Tang, H. Zhang, Leakage identification in water distribution networks with error tolerance capability, Water Resour. Manage. 33 (3) (2019) 1233–1247.
[49] Q. Zhang, Z.Y. Wu, M. Zhao, J. Qi, Leakage zone identification in large-scale water distribution systems using multiclass support vector machines, J. Water Resour. Plann. Manage. 142 (11) (2016) 04016042 1–15..



Marcos Quiñones Grueiro is graduated Summa Cum Laude as an Automation Engineer in 2012 from the Universidad Tecnológica de la Habana "José Antonio Echeverría", CUJAE. He obtained the M.S. degree in Industrial Informatics and Automation in 2017 and the Ph.D. degree in Technical Sciences with specialization in Automation in 2018 from the same university. He has authored journal papers, book chapters, and participated in several international conferences. He has worked on the design of data-driven methods for health monitoring of systems with multiple operating modes. He has developed methods for fault detection and diagnosis of Chemical and Industrial Processes, process safety, and leak detection and location in Water Distribution Networks. His research interests are developing and applying foundational data science methods for the safety, security, and reliability of cyber-physical systems.

Marlon Ares Milán is working towards completing his Automation Engineer degree from the Universidad Tecnológica de La Habana José Antonio Echeverría, CUJAE. His current research interests include fault diagnosis of water distribution networks by using deep learning methods.

Maibeth Sánchez Rivero received her Automation Engineer degree from the Universidad Tecnológica de La Habana "José Antonio Echeverría", CUJAE, Cuba in 2014. She obtained the M.S. degree in Mathematical Modelling applied to Engineering in 2019 in the same University. Her main research interest are in optimization, computational intelligence and fault diagnosis in industrial systems.

Antônio J. Silva Neto is a Full Professor at the Polytechnic Institute of Rio de Janeiro State University, Brazil. He holds Bachelor's and Master's degrees in Mechanical and Nuclear Engineering, respectively, both from the Federal University of Rio de Janeiro (1983, 1989), and a PhD in Mechanical Engineering from North Carolina State University, USA (1993). He is a Titular Member of the Brazilian National Academy of Engineering (ANE), and former president of the Brazilian Scientific Societies on Mechanical Sciences and Engineering (ABCM) and Computational and Applied Mathematics (SBMAC). His main research interests are in heat and mass transfer, numerical methods, optimization, computational intelligence and inverse problems.

Orestes Llanes Santiago received his Electrical Engineer degree from the Universidad Tecnológica de La Habana "José Antonio Echeverría" – CUJAE, Cuba, in 1981. From 1989 to 1994, he pursued graduate studies at the Universidad de Los Andes, Venezuela, where he completed a Master's degree in Control Engineering (1990) and a PhD in Applied Sciences (1994). He is currently a Full Professor and Researcher at the Faculty of Automation and Biomedical Engineering at CUJAE and Titular Member of the Cuban Academy of Sciences. His areas of interest are fault diagnosis in industrial systems, nonlinear control, inverse problems and computational intelligence with applications to control.