Sensor Placement for Classifier-Based Leak Localization in Water Distribution Networks

Adrià Soldevila, Joaquim Blesa, Sebastian Tornil-Sin, Rosa M. Fernandez-Canti, and Vicenç Puig

Abstract This chapter presents a sensor placement method for the classifier-based approaches for leak localization in water distribution networks introduced in the previous chapter. The proposed approach formulates the sensor placement problem as a binary optimization problem. Because of the complexity of the resulting optimization problem, it is solved by means of Genetic Algorithms. In order to reduce the number of sensor configurations to test, a binary matrix that identifies pairs of sensors providing the same information is added as a constraint. The sensors are placed in an optimal way maximizing the accuracy of the leak localization. The proposed approach is first illustrated by means of the application to an academic example based on the Hanoi network. Then, a more realistic case study is proposed based on the Limassol district metered area.

1 Introduction

As already discussed in the previous chapter, leak detection and localization in Water Drinking Networks (WDN) is a subject of major concern for water companies. In the case of complex urban WDN, this is not an easy task to deal with. In order to manage the leak problem and other problems as pressure control, modern urban WDN are usually divided in District Metered Areas (DMA), where the *flow* and the *pressure* at the input are measured [7, 13]. Leakages increase the flow and decrease the pressure measurements at the DMA entrance. However, leak detection and localization are not trivial tasks due to unpredictable variations in consumer demands

---

Adrià Soldevila, Joaquim Blesa, Sebastian Tornil-Sin, Rosa M. Fernandez-Canti, and Vicenç Puig
Research Center for Supervision, Safety and Automatic Control (CS2AC), Rambla Sant Nebridi, s/n, 08022 Terrassa (Spain) e-mail: {adria.soldevila,rosa.mari.fernandez}@upc.edu

Sebastian Tornil-Sin, Joaquim Blesa and Vicenç Puig are also at
Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Carrer Llorens Artigas, 4-6, 08028 Barcelona (Spain) e-mail: {stornil,jblesa,vpuig}@iri.upc.edu

and measurement noise, as well as long-term trends and seasonal effects. Leak detection can be implemented by means of the analysis of the DMA minimum night flow that can also provide an estimation of the leakage level [13]. However, leak localization usually requires the analysis of more than one measured variable and it is a more complex problem. Regarding the type of sensors, although the use of flow measurements is feasible in large water supply networks, this is not the case in WDN where there is a dense mesh of pipes with only flow measurements at the entrance of each DMA. In this case, water companies consider as a feasible approach the possibility of installing some pressure sensors inside the DMAs, because they are cheaper and easier to install and maintain.

In the previous chapter, a classifier-based leak localization architecture and an associated methodology applicable to WDNs is proposed. In a first stage of the proposed architecture, residuals are obtained by comparing available pressure measurements with the estimations provided by a WDN hydraulic model. In a second stage, a classifier is applied to the residuals with the aim of determining the leak location. The classifier is trained with data generated by simulation of the WDN under different leak scenarios and uncertainty conditions. Several classification approaches were considered and compared. As discussed in previous chapter, in the last years, several techniques have been proposed for leak localization purposes such as transient analysis, parameter estimation techniques, leak sensitivity analysis and artificial intelligence methods. Among them, artificial intelligence methods relying on classifiers seem to be a suitable option to deal with the problem of the uncertainty in WDN.

The problem of optimal sensor placement in WDN was first studied for contaminant detection [1, 16]. In recent years, some optimal pressure sensor placement algorithms have been developed to determine which pressure sensors have to be installed inside the DMA such that, with minimum economical costs (number of sensors), a suitable performance regarding leak localization is guaranteed. The main problem of optimal pressure sensor placement is that it leads to a combinatory optimization problem being unfeasible to solve it by evaluating all possible sensor locations. In order to deal with this combinatory problem, the use of Genetic Algorithms (GA) has been proposed by [12] considering a binary leak sensitivity matrix. In [4], GA are used to solve an integer optimization problem based on projections between residuals and the non-binarized leak sensitivity matrix. In [5], the approach of [4] was extended to consider a relaxed isolation index that takes into account an acceptable isolation distance. More recently, the method proposed in [4] has been extended in [18] considering uncertainty.

Alternatively, the use of clustering analysis to group sensors with similar behaviour and reduce the number of combinations to be evaluated is proposed by [15] combined with an efficient branch and bound search. In [3] the model uncertainties are considered in the selection. In [9] the sensor placement has the aim of reducing the isolation error distance. More recently, in [11] the optimal sensor placement problem for the leak localization in WDNs is formulated as a minimum test cover problem.

In this chapter, a sensor placement method for the classifier-based approaches for leak localization in WDNs introduced in previous chapter is presented. Given a number of pressure sensors to be installed in the demand nodes of a DMA, the proposed approach provides the locations of the sensors that maximize the accuracy of a leak localization method that combines the use of pressure models with classifiers (see [17] and previous chapter). The proposed method requires data generated in extensive network simulations. These simulations consider leaks with different magnitudes in all the nodes of the network, differences between the estimated and real consumer water demands, and noise in pressure sensors for all the operating points. Therefore, the presence of model uncertainty is considered in the sensor placement method. Every sensor configuration determines the data that will be used to train the classifier used for the leak localization task. In order to tackle the combinatorial number of sensor configurations to consider, the use of Genetic Algorithms in combination with a sensor distance matrix constraint is proposed to obtain the optimal placement.

2 Background

2.1 Leak Localization using Pressure Residuals and Classifiers

In the previous chapter, an on-line leak localization method based on computing pressure residuals r and analyzing them by a classifier (see Figure 1) is proposed. Residuals are computed as differences between pressure measurements $\tilde{p}$ provided by pressure sensors installed inside the DMA and pressure estimations $\hat{p}_0$ provided by a hydraulic model simulated under leak-free conditions. It is assumed that the network has n_n nodes and that only limited number of sensors can be installed according to budget constraints such that $n_s << n_n$. The WDN model is built using a hydraulic simulator such as Epanet and it is assumed to be able to represent accurately the WDN behavior after the corresponding calibration process using real data. However, it must be noticed that the model is fed with estimated water demands (typically obtained by the total measured DMA demand $\tilde{d}_{WDN}$ and distributed at nodal level according to historical consumption records) in the nodes $(\hat{d}_1, \cdots, \hat{d}_{n_n})$ since in practice real nodal demands $(d_1, \cdots, d_{n_n})$ are not measured (except for some particular consumers where automatic metering readers, AMRs, are available). Hence, the residuals are not only sensitive to leaks but also to the differences between the real demands and their estimated values. Additionally, pressure measurements are subject to the effect of sensor noise $\boldsymbol{\nu}$ and this also affects the residuals. Taking all these effects into account, the classifier must be able to locate the real leak present in the DMA, that can be in any node and with any (unknown) magnitude, while being robust to the demand uncertainty and the measurement noise. Finally, it must be noticed that the operation of the network is constrained by some boundary conditions (for instance, the position of internal valves and reservoir pressures) that are known (measured). These conditions are taken into account in the simulation and can also be used as inputs for the classifier.

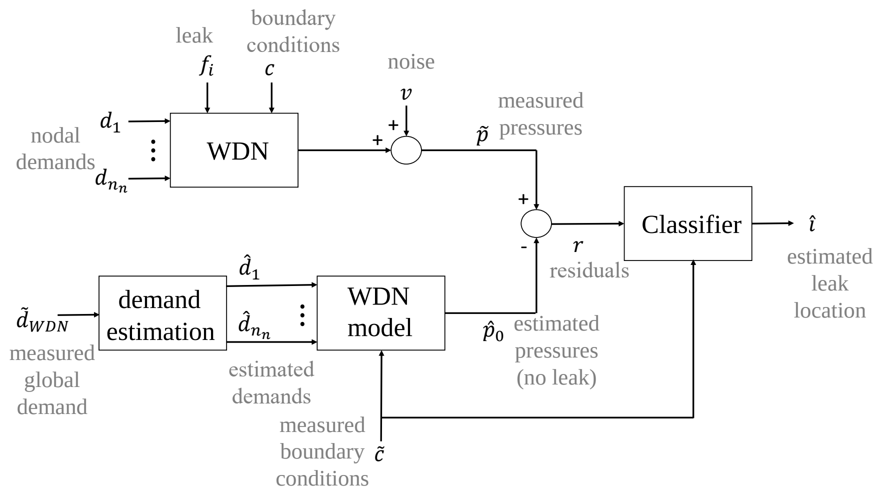

Fig. 1 Leak localization scheme

2.2 Data Generation

The application of the architecture described above (Figure 1) relies on an off-line work whose main goal is to train and validate a classifier able to distinguish the potential leaks under the described uncertainty conditions. In this process, the data generation stage is critical. Since the data that can be obtained from the real monitored WDN can be really limited, the way to obtain a complete training data set is by using the hydraulic simulator. Hence, training (and also validation and testing) data is generated by applying the scheme depicted in Figure 2, similar to the one presented in Figure 1 but with the main difference of substituting the real WDN by a model that allows to simulate the WDN not only in absence but also in presence of faults.

The presented scheme is used to:

- Generate data for all possible leak locations, i.e. for all the different nodes in the WDN ($\bar{f}_i, \quad i = 1, 2, ..., n_n$).
- For each possible leak location, generate data for different leak magnitudes inside a given range ($\bar{f}_i \in [f_i^-, f_i^+]$).
- Generate sequences of demands $(\bar{d}_1, ..., \bar{d}_{n_n})$ and boundary conditions $\hat{c}_i$ that correspond to realistic typical daily evolution in each node.

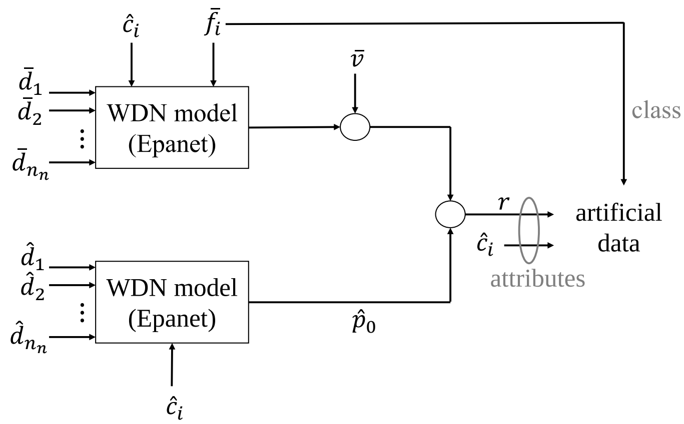

Fig. 2 Data generation scheme

- Simulate differences between the real demands and the estimations computed by the demand estimation module $\left((\bar{d}_1,...,\bar{d}_{n_n}) \neq (\hat{d}_1,...,\hat{d}_{n_n})\right)$.
- Take into account the measurement noise in pressure sensors, by generating synthetic Gaussian noise ($\bar{v}$).

It must be highlighted that the model computes the internal pressures in all the network nodes and that the presented data generation scheme allows generating a complete data set that can be analyzed to determine which pressure measurements are more useful for leak localization purposes.

2.3 Classifier Evaluation

As in the previous chapter, to evaluate the trained classifier for a given sensor configuration, the *confusion matrix* Γ can be computed, which summarizes the results obtained when the classifier is applied to a validation data set. We should recall that when applied to the leak localization problem, the confusion matrix is a square matrix with as many rows and columns as nodes in the network (potential leak locations), where each coefficient $\Gamma_{i,j}$ indicates how many times a leak in node i is recognized as a leak in node j (see Table 1).

In case of a perfect classification, the confusion matrix should be diagonal, with $\Gamma_{i,i} = m$, for all $i = 1, \cdots, n_n$ being m the size of the validation (or testing) data set. In practice, non-zero coefficients will appear outside the main diagonal. For a leak in

Table 1 Confusion matrix Γ

	$\hat{f}_1$	$\cdots$	$\hat{f}_i$	$\cdots$	$\hat{f}_{n_n}$
f_1	$\Gamma_{1,1}$	$\cdots$	$\Gamma_{1,i}$	$\cdots$	Γ_{1,n_n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
f_i	$\Gamma_{i,1}$	$\cdots$	$\Gamma_{i,i}$	$\cdots$	Γ_{i,n_n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
f_{n_n}	$\Gamma_{n_n,1}$	$\cdots$	$\Gamma_{n_n,i}$	$\cdots$	Γ_{n_n,n_n}

node i, the coefficient $\Gamma_{i,i}$ indicates the number of times that the leak $\hat{f}_i$ is correctly identified as $\hat{f}_i$, while $\sum_{j=1}^{n_n} \Gamma_{i,j} - \Gamma_{i,i}$ indicates the number of times that is wrongly classified. The overall accuracy (Ac) of the classifier is defined as:

$$Ac = \frac{\sum_{i=1}^{n_n} \Gamma_{i,i}}{\sum_{i=1}^{n_n} \sum_{j=1}^{n_n} \Gamma_{i,j}}. \tag{1}$$

3 Problem Solution

3.1 Problem Formulation

As already discussed, the objective of this chapter is to develop an approach to place a given number of sensors, n_s, in a DMA of a WDN in order to obtain a sensor configuration with a maximized leak isolability performance when using the leak localization method scheme presented in the previous section. This problem can be recast into the feature selection (also known as variable or attribute selection) problem [19]. The solution of this problem aims at selecting a subset of relevant features (variables, in this case the sensor locations) for use in the classifier construction to maximize their performance. A feature selection algorithm combines a search technique for proposing new feature subsets, along with an evaluation measure which scores the different feature subsets to bring the best subset of features. The simplest algorithm is to test each possible subset of features finding the one which minimizes the error rate. However, this is an exhaustive search of the space that is computationally intractable except for small feature sets.

To select a configuration with n_s sensors, the following binary vector is defined

$$q = \left[\, q_1, \, \cdots \, , \, q_{n_n} \,\right], \tag{2}$$

where $q_i = 1$ if the pressure in the node i is measured, and $q_i = 0$ otherwise (i.e. the vector q denotes which sensors are installed).

In order to evaluate the quality of a sensor configuration regarding its ability to locate a leak at node $i \in \{1, \cdots, n_n\}$, and assuming the case of a single leak, a performance index based on the classified accuracy (1) is proposed.

This performance index depends on the configuration of sensors considered that is parameterized in terms of the binary variable q to determine the best selection

$$Ac(q) = \frac{\sum_{i=1}^{n_n} \Gamma_{i,i}(q)}{\sum_{i=1}^{n_n} \sum_{j=1}^{n_n} \Gamma_{i,j}(q)}. \tag{3}$$

Note that for a given sensor configuration q, $100Ac(q)$ is the percentage of correctly located leaks.

Based on the vector q and the performance index $Ac(q)$ the sensor placement problem can be translated into an optimization problem formulated as follows

$$\max_q Ac(q) \tag{4}$$

s.t.

$$\sum_{i=1}^{n_n} q_i = n_s,$$

where $q \in \{0,1\}$ is defined in (2) and $n_s \in \{1,\ldots,n_n\}$ is the number of sensors that we want to place.

3.2 Sensor Placement using Genetic Algorithms

The optimal sensor placement problem, formulated as the classifier feature selection problem described in previous section, is solved using genetic algorithms and implemented using the Genetic Algorithm (GA) Toolbox of MATLAB.

The overall procedure can be seen in the Figure 3, where the "*Nominal Sensitivity Matrix*" is a data set containing only residuals without uncertainties.

3.2.1 Sensor Distance Matrix

In order to reduce the amount of sensor configurations to be tested in the GA heuristic search, sensor configurations (defined by q) that have at least a pair of sensors with similar behavior in the residual space can be discarded. In order to measure the different behavior of a pair of sensors in the residual space, the leak sensitivity matrix defined in previous chapter can be approximately generated in simulation for a given operating point defined by a nominal network inflow (unique value of water consumption), nominal demand distribution (fixed nodal demand consumption, in this case the $\bar{d}$ is used) and nominal leak size (f^0) [2], approximating the sensitivity components $\Omega_{i,j}$ by

$$\Omega_{i,j}^0 = \frac{\hat{p}_{i,f_j}^0 - \hat{p}_{i,0}^0}{f^0}, \tag{5}$$

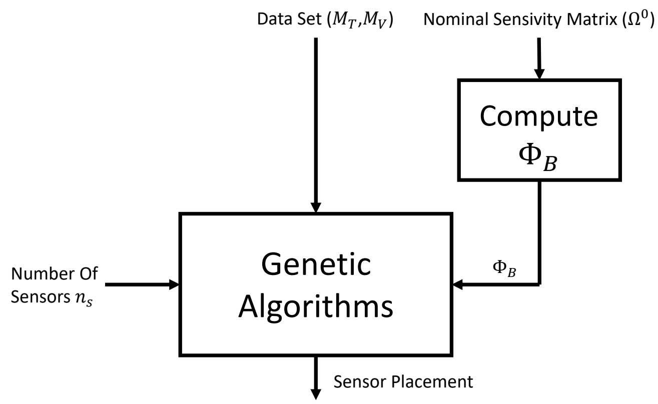

Fig. 3 Scheme of optimization process

where superscript ^{0} denotes the nominal conditions mentioned before.

A criterion that can be used for determining the similarity between sensors is based on comparing the rows of the sensitivity matrix as proposed by [14]. If we consider a nominal approximated sensitivity matrix

$$\Omega^0 = \begin{pmatrix} s_1^0 \\ \vdots \\ s_{n_n}^0 \end{pmatrix}, \tag{6}$$

where s_i^0 $i = 1, ..., n_n$ are row vectors

$$s_i^0 = \left(\Omega_{i,1}^0, ..., \Omega_{i,n_n}^0\right), \tag{7}$$

with components computed using (5), a sensor distance matrix Φ can be defined as

$$\Phi_{i,j} = \|s_i^0 - s_j^0\|_1 \qquad \forall i = 1, ..., n_n \text{ and } j = 1, ..., n_n. \tag{8}$$

Φ is a symmetric square matrix of dimension n_n and diagonal 0. A threshold σ can be determined in order to decide whether two sensors have a different behavior in the residual space or not. Then, a binary matrix $\Phi^{(B)}$ that collects the information of which pairs of sensor combinations are suitable to be in a sensor configuration or not according to their dissimilarity can be computed as

$$\Phi_{i,j}^{(B)} = \begin{cases} 0, \text{ if } \left|\Phi_{i,j}\right| < \sigma \\ 1, \text{ if } \left|\Phi_{i,j}\right| \geq \sigma \end{cases}. \tag{9}$$

3.2.2 Sensor Placement Algorithm

The optimization problem (4) solved by the genetic algorithms has as objective function to be optimized the accuracy defined in (1). The accuracy will be assessed after the classifier training process has ended by using a validation data set as described in Section 2. A training matrix M_T and a validation matrix M_V with data from all the candidate sensors to be installed will be provided to the sensor placement algorithm. For every sensor placement solution, the accuracy obtained using the training and validation data corresponding with the selected sensors will be evaluated.

The training matrix M_T has $n_n + l + 1$ columns where the first n_n columns are the node measurements, the next l columns are the l added attributes (e.g., the measurement of the total water inflow) used by the classifier, and the last column corresponds to the label where each data scenario belongs; the number of rows corresponds to the total number of data scenarios used to train the classifier. The validation matrix M_V has the same column format $(n_n + l + 1)$, and the number of rows corresponds to the total number of data scenarios used for validation purposes.

The pseudo-code of the algorithm is shown in Algorithm 1. First, we initialize the variables of the GA (line 1) including the bit string type population, the tolerance, the population size p and the elite count in order to save part of the previous analyzed results. Then, we declare the search constraints (line 2) being n_s the constraint of the set of possible solutions for each variable and the number of sensors. Then, in the optimization process (lines 4 to 24), an initial matrix with random sensor positions is delivered by the GA (line 6). Before to proceed with the objective function optimization, it is checked (with the function GetUsed() in line 9) if the sensor configuration has already been considered (as proposed by [10]). The stored value is retrieved with the function GetAc (line 20). If not yet considered, the sensor configuration is considered to be tested. If the new sensor configuration is not tested, and if all the sensor pairs are suitable to be in a sensor configuration according the binary sensor distance matrix (9) and the function CheckCombinations() (which returns 1 if the configuration is allowed (line 10)), the sensor placement configuration is tested evaluating the objective function (line 13) and the combination is stored as used with the function SetUsed (line 14). Then, the Ac value obtained is also stored with the function SetAc (line 15). If there is at least one forbidden pair of sensors in the sensor configuration, the configuration is discarded and a zero value is assigned to the objective function (line 17). The binary vector q allows the selection of the adequate columns of the matrices M_T and M_V in order to train (line 11), validate (line 12) and compute Ac (line 13) for the classifier according the selected nodes to be measured. Once the Ac value has been obtained for all members of the matrix I, we look for the maximum value (line 23). Then, the optimization is finished and the sensor placement selected is the one that provide the best Ac value.

Algorithm 1 Sensor placement based on Genetic Algorithms

Require: A training matrix M_T and a validation matrix M_V. The number of features to select n_s, the number of nodes n_n, the population size p and the binarized matrix $\Phi^{(B)}$.

Ensure: A near-optimal sensors configuration $\mathbf{q}$ with error index Ac_{max}.

$init \leftarrow$ InitVarGA()
$constraint \leftarrow$ SetConstraints()
Inputs: $init$, $constraint$, M_T, M_V, p, n_n, $\Phi^{(B)}$.
while An optimization criterion is not reached **do**
    GA based search:
    Generate I matrix of size $(p \times n_n)$ where each row is a member of a generation.
    for $k = 1, \cdots, p$ **do**
        $q(k) \leftarrow I(k)$
        if GetUsed($\mathbf{q}(k)$) $= 0$ **then**
            if CheckCombinations($\Phi^{(B)}, q(k)$) $= 1$ **then**
                $C(q(k)) \leftarrow$ Train($M_T(q(k))$)
                $\Gamma(\mathbf{q}(k)) \leftarrow$ Validate($C(q(k)), M_V(q(k))$)
                $Ac(q(k)) \leftarrow \frac{\sum_{i=1}^{n_n} \Gamma_{ii}(q(k))}{\sum_{i=1}^{n_n} \sum_{j=1}^{n_n} \Gamma_{ij}(q(k))}$
                SetUsed($q(k)$)
                SetAc($Ac(q(k), q(k)$)
            else
                $Ac(q(k)) = 0$
            end if
        else
            $Ac(q(k)) =$ GetAc($q(k)$)
        end if
    end for
    Find $\{q, Ac_{max}\}$ such that $Ac_{max} = \max_q (Ac(q(1), ..., Ac(q(p)))$.
end while

4 Case Studies and Results

The proposed sensor placement approach is tested in two different networks. On the one hand, a small size network (Hanoi) is used since it allows to compare the proposed approach to the results obtained using the exhaustive search method. On the other hand, a medium size network (Limassol) shows the performance in a more realistic scenario.

All the results have been obtained using a PC with an INTEL(R) CORE(TM) i7-4720HQ CPU @ 2.60 [GHz], 8 [GB] of memory RAM, a Windows 10 Home 64 bits operative system and using the MATLAB 2015a software [8]. The sensor placement approach follows Algorithm 1 and uses MATLAB GA Toolbox that, for the considered case studies, has been used considering the following parameters:

- Tolerance of 10^{-6}.
- Population size $p = 5$.
- Elite count of $0.05p$, but at least one survives (which is the case given the p selected).
- The maximum number of generations is 50.

4.1 Hanoi Case Study

The Hanoi (Vietnam) network, presented in Figure 4, is a simplified network of the real one, and consists of 1 reservoir, 31 consumer nodes and 36 pipes. The water consumption has a daily pattern similar to the one depicted in Figure 5 (all the water consumption patterns have been generated from an unique pattern distribution obtained from the average values of five days adding an uncertainty of $\pm$ 12.5 %). Given the size of the network, it is considered the placement of only two pressure sensors as presented by [4]. These sensors, and the flow sensor at the inlet are considered to operate with a sampling time of 10 minutes.

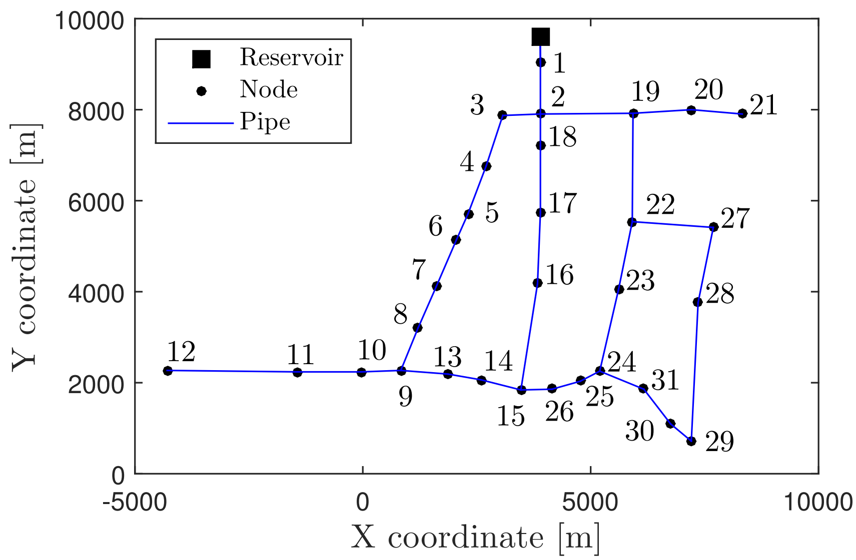

Fig. 4 Hanoi topological network

To generate the data sets, three different uncertainty sources (as proposed by [6]) are considered in the following way:

- The demand uncertainty source has a magnitude of $\pm$ 10 % of the nominal node consumption value.
- The leak size varies from 25 to 75 [l/s].
- The measurement noise magnitude is considered as the of $\pm$ 5 % of the average value of all pressure residuals.

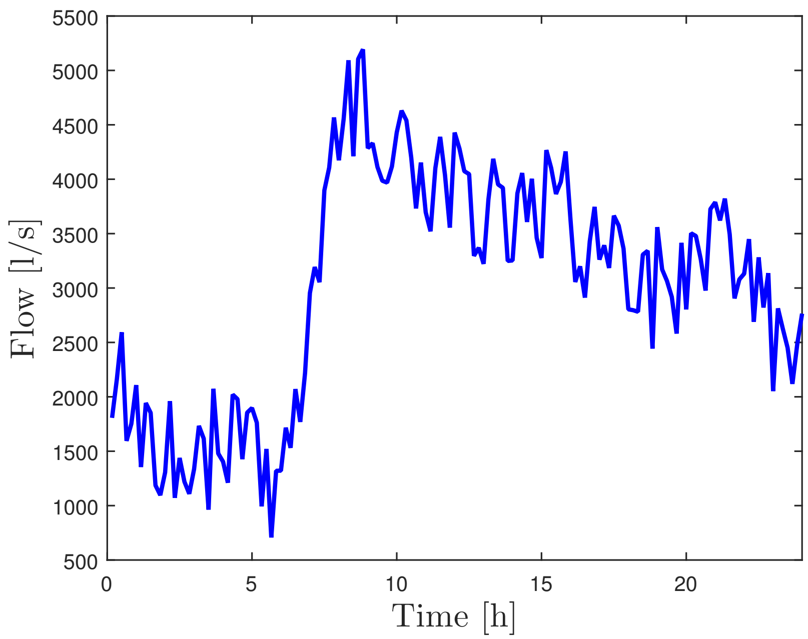

Fig. 5 Example of a daily flow consumption in the Hanoi network

Using all of these uncertainty levels, ten complete data sets (in order to have a large enough number of different scenarios to avoid outliers or strange cases) have been created simulating the pressure measurements at each possible position (class) where data are generated every ten minutes. Then, the hourly average value has been computed (with the aim to reduce the uncertainty and remove outliers). Thus, each complete data set is composed of a training data set with five days of data (120 samples for each class) and a validation data set with one day of data (24 samples for each class). Finally a unique testing data set with ten days of data (240 samples per class) is generated. For the normalized sensitivity matrix (5), one instance is generated for each class and sensor (complete sensitivity matrix) with a value of total consumption of water of 2991.1 $\mathrm{l/s}$ and leak size of 50 $\mathrm{l/s}$.

Classifiers use as attributes the flow measurement at the inlet, and the two pressure residuals from the node where the sensor configuration is assessed. The proposed sensor placement method using GA and with/without the $\Phi^{(B)}$ matrix (where it is used a σ value of the average value of all the Φ except the diagonal) is compared to the exhaustive search. The results for the k-NN classifier (for a k value equal to one, since the election of a proper k value must be done when the sensor placement is fixed) are summarized in Table 2, and for the case of the Bayesian classifier (with variable dependent Gaussian PDFs) in Table 3. The genetic algorithms are designed to store only the best member of each generation, and each generation (population size) is fixed to have five members. To compute the Ac value in both tables, the same testing data set is computed for each sensor placement obtained (with their respective training data set). The time units in the tables are seconds, and the Ac values are in %. The best configurations (highest accuracy performance over the testing data set) obtained are highlighted in bold for each method.

Table 2 Sensor placement results in the Hanoi network for the k-NN classifier

Data set	Exhaustive search			Genetic algorithm			Genetic algorithm + $\Phi^{(B)}$			
	Sensors	Time	Ac	Sensors	Time	Ac	Sensors	Time filter	Time GA	Ac
1	14, 27	474	39.11	9, 15	71	31.19	14, 27	0.06	63	39.11
2	14, 29	471	38.14	14, 29	70	38.14	14, 29	0.06	38	38.68
3	14, 28	470	38.68	14, 28	47	38.68	14, 31	0.06	19	34.34
4	**14, 28**	**499**	**41.06**	**14, 28**	**44**	**41.06**	**14, 28**	**0.06**	**33**	**41.06**
5	14, 28	472	39.03	1, 30	14	16.80	14, 28	0.06	46	39.03
6	14, 27	473	38.02	10, 15	45	32.58	14, 27	0.06	36	36.07
7	15, 28	473	38.52	15, 28	55	38.52	15, 28	0.06	43	38.52
8	15, 28	477	38.02	26, 28	29	35.55	15, 28	0.06	23	38.02
9	14, 27	469	38.89	5, 14	50	31.16	14, 29	0.06	36	37.56
10	14, 28	474	38.91	15, 29	51	36.88	4, 15	0.06	15	29.04
Average	-	475	38.83	-	47	34.05	-	0.06	35	37.14

Table 3 Sensor placement results in the Hanoi network for the Bayesian classifier

Data set	Exhaustive search			Genetic algorithm			Genetic algorithm + $\Phi^{(B)}$			
	Sensors	Time	Ac	Sensors	Time	Ac	Sensors	Time filter	Time GA	Ac
1	14, 28	537	51.57	9, 15	71	42.00	14, 28	0.06	47	51.57
2	14, 28	535	51.65	14, 29	69	51.65	14, 28	0.06	57	51.65
3	14, 28	544	52.01	14, 28	47	52.01	14, 28	0.06	55	52.01
4	**14, 28**	**545**	**52.55**	**14, 28**	**44**	**52.55**	**14, 28**	**0.06**	**51**	**52.55**
5	14, 28	578	51.41	1, 30	13	46.47	4, 13	0.06	29	35.53
6	26, 27	583	46.72	10, 15	44	43.99	7, 28	0.06	32	40.13
7	14, 28	537	52.12	14, 28	85	52.12	14, 29	0.06	72	50.73
8	13, 28	536	47.33	13, 28	55	47.33	13, 28	0.06	62	47.33
9	14, 28	599	52.37	5, 14	50	43.56	14, 28	0.06	53	52.37
10	15, 28	535	46.92	6, 26	40	37.29	15, 30	0.06	50	46.57
Average	-	553	50.46	-	58	46.90	-	0.06	51	48.04

From these results, it can be seen that both methods present an important improvement in terms of computational time when GA are used, and the GA standalone method and GA plus $\Phi^{(B)}$ are able to avoid the local minima and find the global optima in some cases (the best sensor placements obtained). Moreover, notice that in average the introduction of the $\Phi^{(B)}$ matrix not only reduces significantly the computational time compared to the purely GA method but also increases the accuracy. Finally, compared to the k-NN classifier, the Bayesian classifier is more time demanding but its accuracy is better. This is probably due to the fact that the allowed combinations are better (i.e., the criteria used to select the permitted pairs of sensor configurations works better) for this classifier than for the k-NN classifier.

To decide the best sensor configuration, the one with highest accuracy value is chosen. So, for the technique of the genetic algorithms plus the use of $\Phi^{(B)}$ matrix, in case of k-NN classifier, the best sensor placement obtained is at nodes 14 and 28. On the other hand, for the Bayesian classifier case, the best sensor placement is also at the nodes 14 and 28. In both cases, the accuracy is assessed using a time horizon scheme (see previous Chapter) in Figure 6 for the k-NN classifier (with $k = 1$) and in Figure 7 for the Bayesian classifier, both using the training data corresponding to the first data set and using as testing data set of all the remaining data sets. The term "node relaxation" refers to the number of nodes in topological distance between the node with the real leak and the node where the classifier predicts the leak for which the diagnosis is still considered correct.

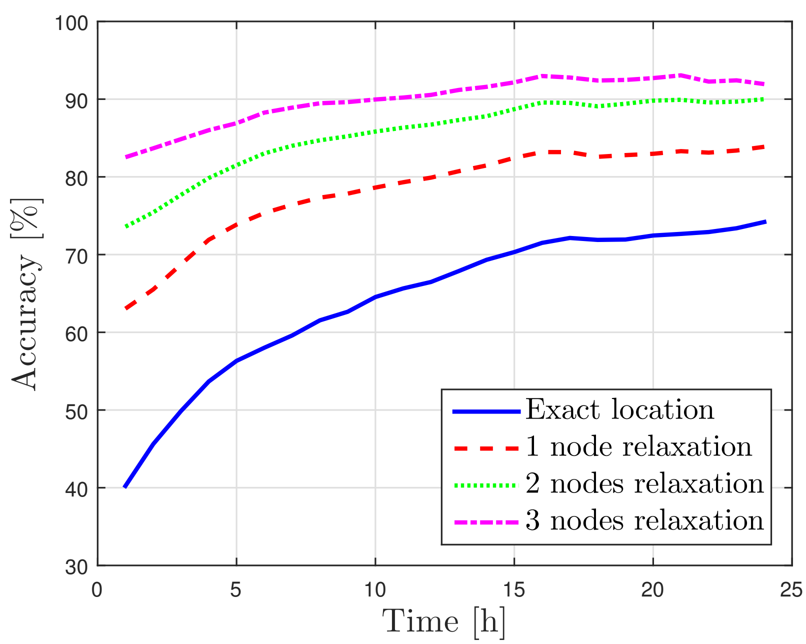

Fig. 6 Accuracy curves for the k-NN classifier in the Hanoi network with sensor placement at nodes 14 and 28

The average topological distance, which is the average value of the minimum distance in nodes between the node predicted by the classifier and the real node with leak, is depicted in Figure 8 for both classifiers.

The results in this network show that the best performance is achieved with the Bayesian classifier being in agreement with the results presented in previous chapter. The sensor placement result can be seen in Figure 9.

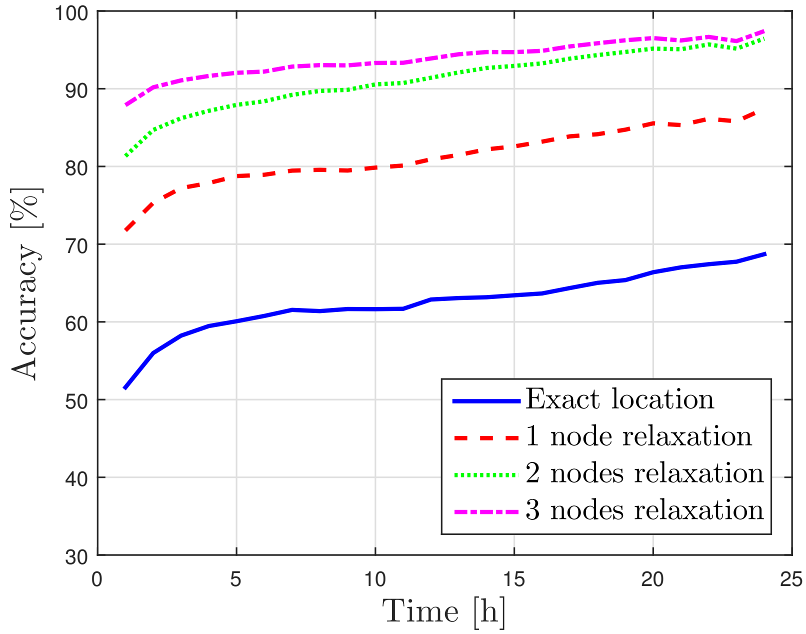

Fig. 7 Accuracy curves for the Bayesian classifier in the Hanoi network with sensor placement at nodes 14 and 28

4.2 Limassol Case Study

The Limassol (Cyprus) network (presented in Figure 10) has a medium size with 1 reservoir, 197 consumer nodes and 236 pipes. The consumption of water has a pattern (depicted in Figure 11) generated as in the previous case (with different scale). For this network, it is decided to place three pressure sensors.

The data sets are generated considering similar uncertainties (in this case the leak varies from 2 to 6 [l/s]) as the Hanoi case, same sampling time (and computing the hourly average value) and the same number of examples for each class. The classifiers are build like the Hanoi case, but using four attributes: the flow measurement at the inlet, and the three pressure residuals. For the sensitivity matrix the considered values are 492.2 [l/s] for the total water consumption, and 4 [l/s] for the leak size.

The results are summarized in the Table 4.2 for the case of the k-NN classifier ($k = 1$ as in the Hanoi case), and in the Table 4.2 for the Bayesian classifier. In this case, the σ value is equal to the mean values of the Φ matrix except the diagonal. The best configurations are highlighted in bold for each method.

In this network, as in the Hanoi case, the use of the $\Phi^{(B)}$ matrix reduces the computation time in most cases and in the average value. In this network, the computational time reduction is similar, but the degradation of the accuracy value is

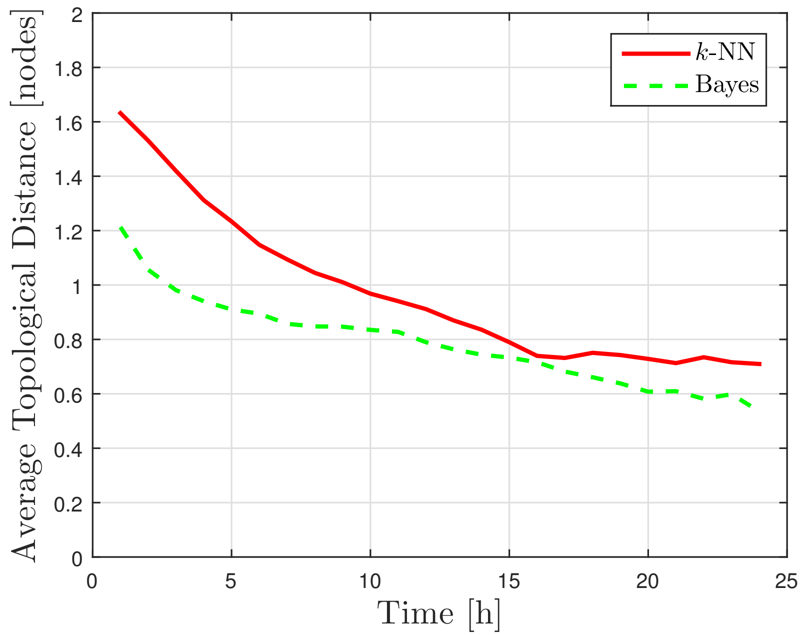

Fig. 8 Average topological distance for both classifiers classifier in the Hanoi network

Table 4 Sensor placement results in the Limassol network for the *k*-NN classifier

Data set	Genetic algorithm			Genetic algorithm + $\Phi^{(B)}$			
	Sensors	Time	Ac	Sensors	Time filter	Time GA	Ac
1	**15, 46, 113**	**8628**	**11.03**	1, 7, 195	6.4	7348	9.04
2	1, 7, 11	8602	10.89	8, 102, 182	7.5	16154	10.19
3	8, 183, 195	14906	9.84	52, 128, 133	6.8	8020	9.66
4	124, 183, 185	13957	8.09	7, 195, 197	6.2	6580	9.66
5	3, 7, 8	5508	9.66	1, 7, 195	7.1	1595	8.80
6	6, 8, 11	13652	10.72	104, 183, 195	6.8	5909	8.36
7	129, 185, 190	15308	8.03	1, 2, 197	6.3	479	5.38
8	1, 3, 7	2010	9.07	13, 40, 167	6.3	8294	10.37
9	87, 124, 128	12434	9.55	104, 124, 167	8.0	10551	10.37
10	3, 166, 181	4995	7.11	**5, 11, 124**	**6.2**	**13304**	**10.39**
Average	-	9690	9.39	-	6.7	7793	9.22

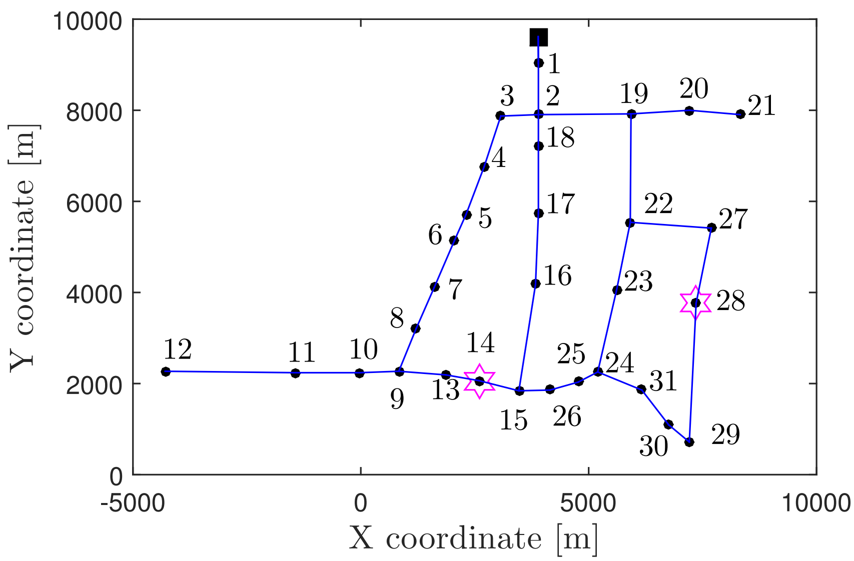

Fig. 9 Sensor placement for the *k*-NN and Bayesian classifiers in Hanoi network

Table 5 Sensor placement results in the Limassol network for the Bayesian classifier

Data set	Genetic algorithm			Genetic algorithm + $\Phi^{(B)}$			
	Sensors	Time	Ac	Sensors	Time filter	Time GA	Ac
1	39, 77, 133	21348	17.35	11, 46, 133	6.3	21255	18.90
2	7, 19, 23	18701	19.28	17, 166, 181	6.3	17845	19.60
3	45, 110, 185	16696	16.40	40, 75, 156	6.8	14742	15.74
4	7, 11, 110	47590	19.34	**17, 46, 181**	**6.7**	**25515**	**19.63**
5	11, 91, 186	28032	17.43	91, 188, 190	6.5	20117	15.67
6	39, 48, 485	28106	19.13	39, 93, 189	6.7	15775	17.23
7	94, 133, 166	7730	19.22	100, 124, 183	6.9	15765	18.90
8	124, 189, 192	29217	16.61	14, 167, 185	6.4	13200	17.93
9	13, 22, 100	21904	18.86	93, 188, 190	6.4	11453	15.97
10	**40, 66, 104**	**15221**	**20.10**	13, 22, 190	6.5	6416	17.63
Average	-	23454	18.37	-	6.5	16208	17.72

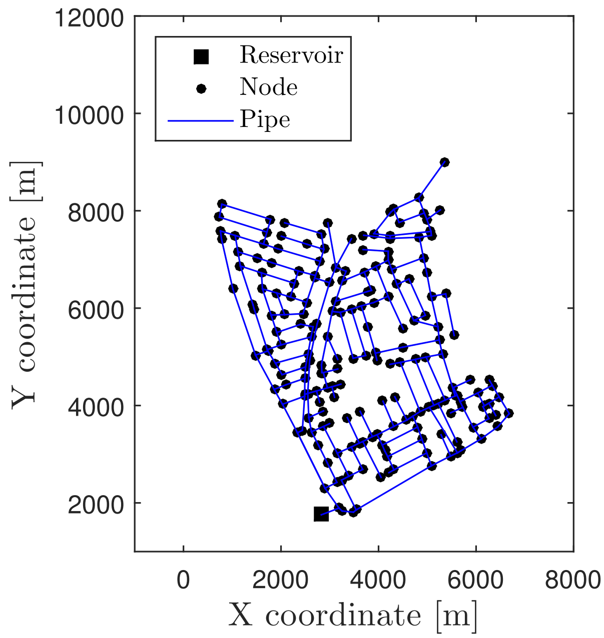

Fig. 10 Limassol network

worse. This is probably due to the low value of the population size compared with the total number of combinations possible. The best result obtained for the k-NN classifier is to place the sensors at nodes 5, 11 and 124, and for the Bayesian classifier is to place the sensors at nodes 17, 46 and 181. The accuracy curves (using the first training data set and the testing data set) for both sensor placements is depicted in Figure 12 and Figure 13, respectively.

The average topological distance for k-NN and Bayesian classifiers for the sensor placements obtained can be seen in Figure 14.

Finally, the resulting sensor placement for the last proposed method (Genetic algorithm + $\Phi^{(B)}$) for both classifiers is depicted in Figure 15.

5 Conclusions

In this paper, an optimal sensor placement method for placing a given number n_s of pressure sensors in WDNs to be used for leak localization has been presented. The obtained sensor configuration is optimal in the sense that it maximizes the leak isolability when using a classifier-based leak localization method. In order to tackle the complexity of the optimal sensor placement, the use of Genetic Algorithms and

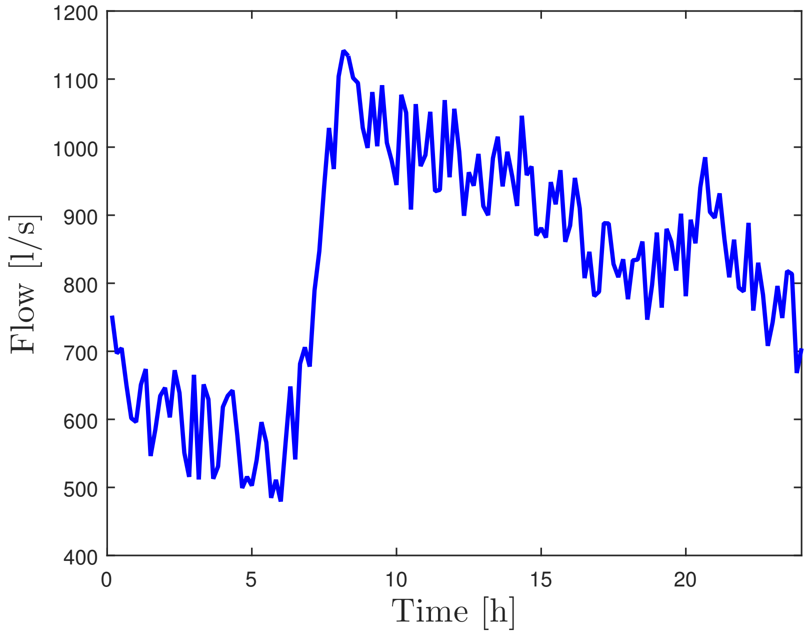

Fig. 11 Example of a daily flow consumption in the Limassol network

a method to reduce that number of sensor configurations to test has been proposed. The performance of the proposed method has been illustrated by means of the application to the Hanoi and Limassol networks. The simulation results show that the Genetic Algorithm provides in average a similar performance index to the Exhaustive Search Algorithm whereas the computation load decreases significantly. Thus, this method becomes suitable for networks of growing complexity. Additionally, the use of classifiers allows the direct introduction of different sources of uncertainty and leads to good isolability results.

6 Acknowledgment

This work has been partially funded by the Spanish Government (MINECO) through the project CICYT ECOCIS (ref. DPI2013-48243-C2-1-R), by MINECO and FEDER through the project CICYT HARCRICS (ref. DPI2014-58104-R), by MINECO through the grant IJCI-2014-2081, by EFFINET grant FP7-ICT-2012-318556 of the European Commission, by the DGR of Generalitat de Catalunya (SAC group Ref. 2014/SGR/374) and by the AGAUR of Generalitat de Catalunya through the grants FI-DGR 2015 (ref. 2015 FI_B 00591) and 2014PDJ00102.

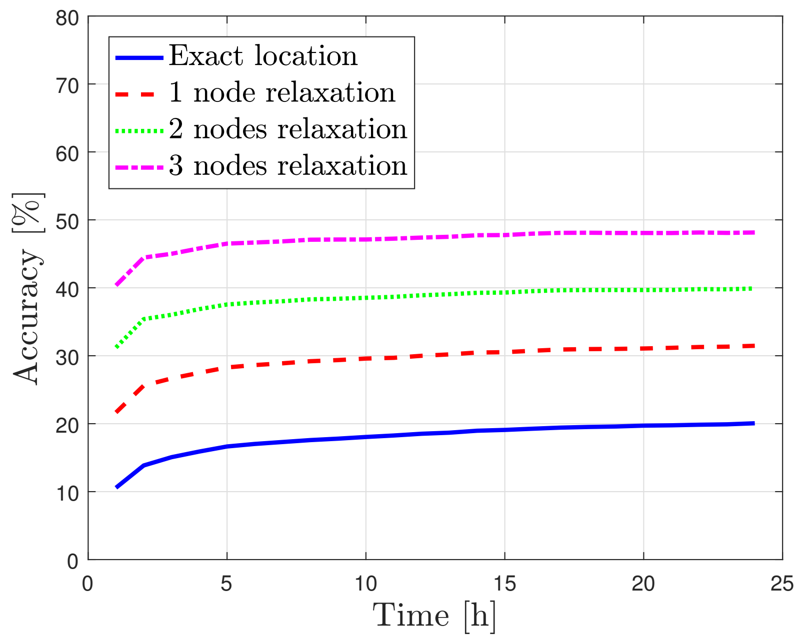

Fig. 12 Accuracy curves for the *k*-NN classifier in the Limassol network with sensor placement at nodes 5, 11 and 124

References

1. Berry, J., Fleischer, L., Hart, W., Phillips, C.: Sensor placement in municipal water networks. pp. 3383–3392 (2003)
2. Blesa, J., Nejjari, F., Sarrate, R.: Robustness analysis of sensor placement for leak detection and location under uncertain operating conditions. Procedia Engineering **89**, 1553–1560 (2014)
3. Blesa, J., Nejjari, F., Sarrate, R.: Robust sensor placement for leak location: Analysis and design. Journal of Hydroinformatics **18**(1), 136–148 (2016)
4. Casillas, M.V., Puig, V., Garza-Castañón, L.E., Rosich, A.: Optimal Sensor Placement for Leak Location in Water Distribution Networks Using Genetic Algorithms. Sensors **13**(11), 14,984–15,005 (2013)
5. Cugueró-Escofet, M.À., Puig, V., Quevedo, J., Blesa, J.: Optimal pressure sensor placement for leak localisation using a relaxed isolation index: Application to the Barcelona water network. IFAC-PapersOnLine **48**(21), 1108 – 1113 (2015)
6. Cugueró-Escofet, P., Blesa, J., Pérez, R., Cugueró-Escofet, M.À., Sanz, G.: Assessment of a leak localization algorithm in water networks under demand uncertainty. IFAC-PapersOnLine **48**(21), 226 – 231 (2015)
7. Lambert, M.F., Simpson, A.R., Vítkovský, J.P., Wang, X.J., Lee, P.J.: A review of leading-edge leak detection techniques for water distribution systems. In: 20th AWA Convention, Perth, Australia (2003)
8. The Mathworks, Inc., Natick, Massachusetts: MATLAB R2015a version (2015)
9. Nejjari, F., Sarrate, R., Blesa, J.: Optimal pressure sensor placement in water distribution networks minimizing leak location uncertainty. Procedia Engineering **119**(1), 953–962 (2015)

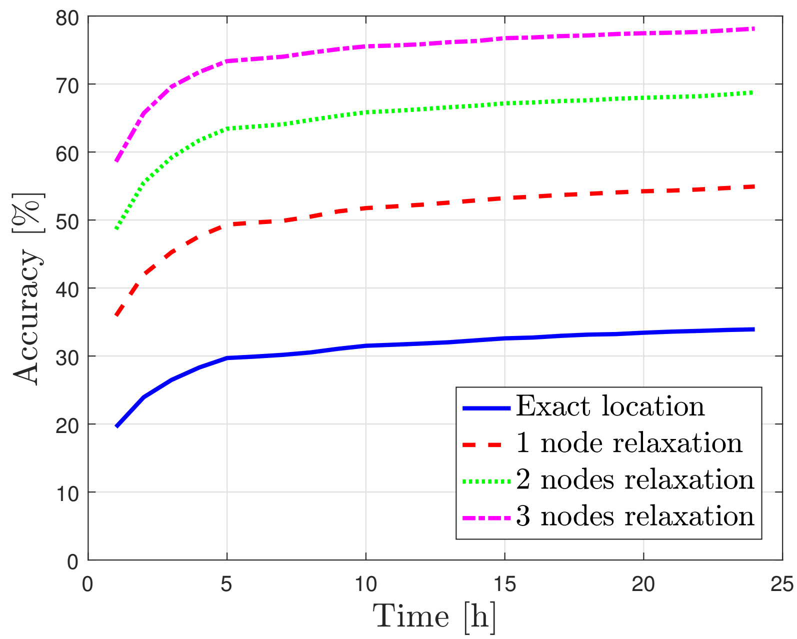

Fig. 13 Accuracy curves for the Bayesian classifier in the Limassol network with sensor placement at nodes 17, 46 and 181

10. Oh, I.S., Lee, J.S., Moon, B.R.: Hybrid genetic algorithms for feature selection. IEEE transactions on pattern analysis and machine intelligence **26**(11), 1424–1437 (2004)
11. Perelman, L.S., Abbas, W., Koutsoukos, X., Amin, S.: Sensor placement for fault location identification in water networks: A minimum test cover approach. Automatica **72**, 166 – 176 (2016)
12. Pérez, R., Puig, V., Pascual, J., Peralta, A., Landeros, E., Jordanas, L.: Pressure sensor distribution for leak detection in Barcelona water distribution network. Water Science and Technology: Water Supply **9**(6), 715–721 (2009)
13. Puust, R., Kapelan, Z., Savić, D.A., Koppel, T.: A review of methods for leakage management in pipe networks. Urban Water Journal **7**(1), 25–45 (2010)
14. Sarrate, R., Blesa, J., Nejjari, F.: Clustering techniques applied to sensor placement for leak detection and location in water distribution networks. In: Control and Automation (MED), 2014 22nd Mediterranean Conference of, pp. 109–114 (2014)
15. Sarrate, R., Blesa, J., Nejjari, F., Quevedo, J.: Sensor placement for leak detection and location in water distribution networks. Water Science and Technology: Water Supply **14**(5), 795–803 (2014)
16. Shastri, Y., Diwekar, U.: Optimal sensor placement for water distribution network security. pp. 6713–6736 (2005)
17. Soldevila, A., Blesa, J., Tornil-Sin, S., Duviella, E., Fernandez-Canti, R.M., Puig, V.: Leak localization in water distribution networks using a mixed model-based/data-driven approach. Control Engineering Practice **55**, 162–173 (2016)
18. Steffelbauer, D.B., Fuchs-Hanusch, D.: Efficient sensor placement for leak localization considering uncertainties. Water Resources Management pp. 1–17 (2016)

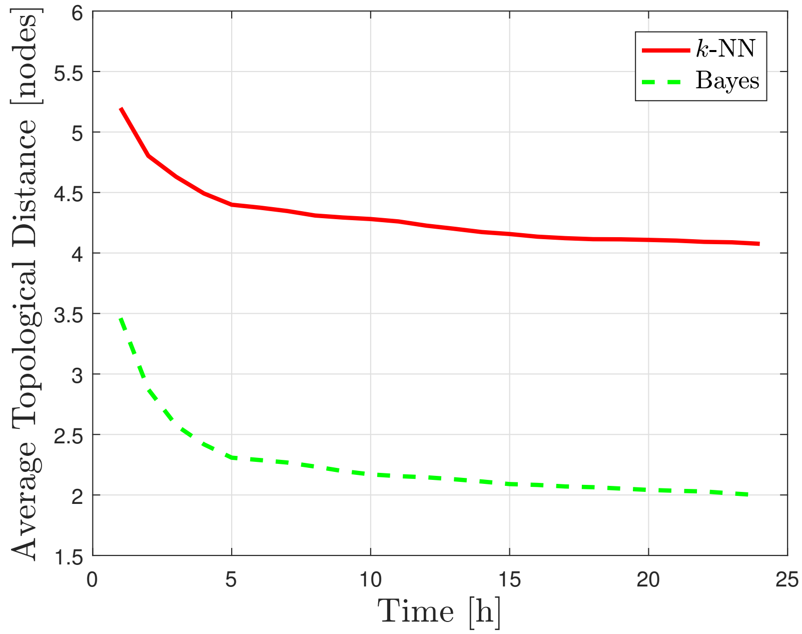

Fig. 14 Average topological distance for both classifiers in the Limassol network

19. Tang, J., Alelyani, S., Liu, H.: Feature selection for classification: A review. In: C. Aggarwal (ed.) Data Classification. Chapman and Hall/CRC

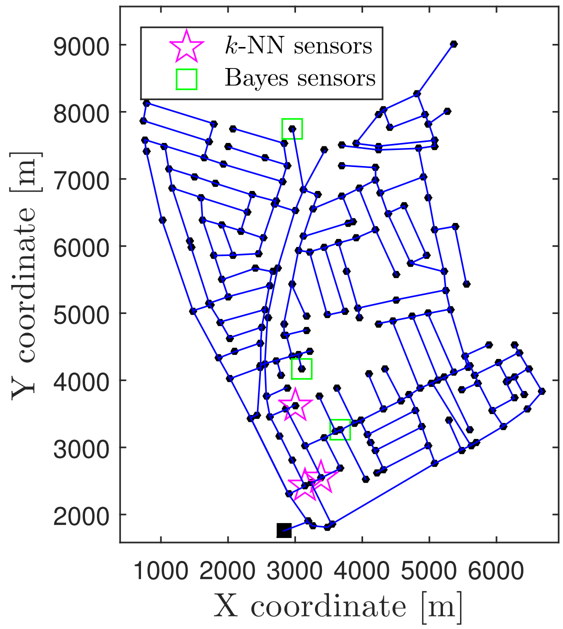

Fig. 15 Sensor placement for the k-NN and Bayesian classifiers in the Limassol network