Chapter 6

Leak localization in water distribution networks using machine learning based on cosine features

Ildeberto Santos-Ruiz[1], Francisco-Ronay López-Estrada[1], Vicenç Puig[2], and Guillermo Valencia-Palomo[3]

Abstract Locating leaks in water distribution networks is vital in drinking water management. This chapter presents a method for leak localization using pressure measurements modified by a nonlinear transformation related to direction cosines in the pressure space. Direction cosines are used as input variables in classifiers that locate leaks by associating them to the closest node in the distribution network. The feature extraction process is derived from hyper-parameter optimization with k-NN classifiers, but it is generalized to other classifiers based on supervised learning. The method was tested with physical measurements in some sectors of the Madrid network and synthetic data obtained by simulation with EPANET on a model of the Hanoi network. Some comparisons with other machine learning techniques, using raw pressures and pressure residuals, are performed to illustrate leak localization effectiveness when considering directional features.

6.1 Introduction

Loss caused by leakage is an underlying problem in drinking water management, as about one third of chemically treated water is lost globally because of leaks in distribution systems (OECD, 2016). Leaks must be detected and located promptly to be repaired in a short time and minimize the volume leaked. The leak localization problem becomes challenging because many leaks cannot be located visually because of the leaking liquid seeping underground instead of emerging to the surface.

The leak management process in a Water Distribution Network (WDN) consists of several stages. First, the leak must be perceived: the existence of water loss must be detected. If the leak is not visible, then a small region enclosing the leak (a leakage hotspot) must be estimated using calculations based on available hydraulic measurements. Second, to repair a leak, it must be accurately located using physical inspection and specialized devices, e.g., a geophone and correlators (Pilcher et al., 2007; Puig et al., 2017). The methods described in this chapter belong more to the prelocalization stage, which is intended

[1]Tecnológico Nacional de México / I.T. Tuxtla Gutiérrez, Carretera Panamericana S/N, 29050 Tuxtla Gutiérrez, Mexico,
e-mail: ildeberto.dr@tuxtla.tecnm.mx and frlopez@tuxtla.tecnm.mx

[2]Department of Automatic Control (ESAII), Universitat Politècnica de Catalunya UPC, Rambla de Sant Nebridi 10, 08222 Terrassa, Spain
e-mail: vicenc.puig@upc.edu

[3]Tecnológico Nacional de México, IT Hermosillo, Av. Tec. y Per. Poniente S/N, Hermosillo 83170, Mexico,
e-mail: gvalencia@hermosillo.tecnm.mx

to locate the closest nodes to the leak. Determining these nodes is helpful for the operators to locate the actual leak point in a short time. This task, however, is not trivial because of the following: small number of sensor measurements available compared with a large number of possible leak locations, the uncertainty in individual user consumption, and the noise and unpredictable transients that affect the certainty of the measurements. These difficulties, among others, make locating leaks a challenging problem; even with the results and achievements that will be discussed in the final sections, the leak localization problem remains open.

The proposed method is based on pressure residuals, which are the difference between nominal pressures (leak-free pressures, estimated from a network model) and current leaky pressures measured at specific network sensing nodes. A substantial difference from other leak localization methods is that the proposed method is not based on the magnitude of the residuals but on its direction, which is characterized by the so-called direction cosines.

The chapter first presents the state of the art that summarizes the most relevant previous works in this area. A simplifying hypothesis in these works is that single leaks occur at the nodes. Although leaks can appear anywhere in the network, this assumption simplifies the computation to estimate their position.

The leak localization problem is generally undetermined. Therefore, localization methods may fail to isolate the correct leaky node. It is possible, however, to identify candidate leaky nodes close to the actual leaky node, and it is essential to determine how good the predictability of each method is. Therefore, after presenting the literature review, some metrics will be described to evaluate a specific leak localization method and its effectiveness compared with other methods. The case studies used to validate the proposed methods will then be presented. Next, the methodology for leak localization considering both time-independent (steady-state) and extended period leak localization will be described. Finally, the results obtained in the case studies will be presented and discussed.

6.2 State of the art on leak localization in WDN

In the literature, there are different works reported in the context of leak detection and localization; for example, Pudar and Liggett (1992) formulated a solution based as an inverse problem of a steady-state hydraulic model by minimizing the differences between measurements and simulated states. By solving the inverse problem, the network model parameters, such as leak flow rate and location, are determined from the sensor measurements. However, the inverse problem is generally undetermined in an operational water network because of the small number of sensor measurements available.

To solve the indeterminate problem for leak localization when there are multiple equivalent leak candidates, Pudar and Liggett (1992) minimized the L_2-norm of the leak parameters. Conversely, Berglund et al. (2017) reduced the number of unknown measurements by selecting candidate nodes before solving the optimization problem. Recently, Blocher et al. (2020) addressed the inverse problem of leak localization using regularization methods to mitigate the ill-posedness effect. Using a regularization term, the authors solve the inverse problem to estimate leak hotspots by minimizing the sum of squared residuals, which represent the difference between measurements and steady-state model simulations. The performance of leak localization with this method depends on the regularization parameter choice and does not consider the uncertainty in pressures and residuals.

Sanz and Pérez (2015) and Sanz et al. (2016) also formulated an underdetermined inverse problem for demand calibration in order to solve the leak localization problem. To solve the ill-posed problem, Sanz and Pérez (2015) grouped the demand nodes based on the measured pressure sensitivity to a change in a node demand. The deviation of the calibrated demand from the expected demand for a given group indicates the presence and approximate location of a leak (Sanz et al., 2016). Prior assumptions about leak candidates and node groups are required, affecting the leak location search.

An alternative approach, which avoids the ill-posed inverse problem, directly compares the residuals with the pressure sensitivities for different leak locations (Casillas et al., 2013). Although this approach based on the sensitivities is limited to single leak scenarios, it has shown promising results in real applications (Pérez et al., 2011; Pérez et al., 2014).

The *pressure sensitivity matrix*, $\mathbf{S}$, expresses how sensitive the pressure at each node is to a leak occurring at any other node (even at the same node). Thus, the element s_{ij} that relates the leak flow rate q_i^ℓ at the i-th node with the pressure P_j at the j-th node is defined as $\frac{\partial P_j}{\partial q_i^\ell}$ so that

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_N \end{bmatrix} = \begin{bmatrix} \partial P_1/\partial q_1^\ell & \partial P_2/\partial q_1^\ell & \cdots & \partial P_N/\partial q_1^\ell \\ \partial P_1/\partial q_2^\ell & \partial P_2/\partial q_2^\ell & \cdots & \partial P_N/\partial q_2^\ell \\ \vdots & \vdots & \ddots & \vdots \\ \partial P_1/\partial q_N^\ell & \partial P_2/\partial q_N^\ell & \cdots & \partial P_N/\partial q_N^\ell \end{bmatrix}, \tag{6.1}$$

where N is the number of nodes in the network and $\mathbf{s}_i$ is the vector of sensitivities to the leak at the i-th node.[1] To locate the leak, each sensitivity vector $\mathbf{s}_i$ is compared with the residual vector at each time. The residual vector, $\mathbf{r}$, is defined by

$$\mathbf{r} = [P_1 - P_1^{\text{nom}}, P_2 - P_2^{\text{nom}}, \ldots, P_N - P_N^{\text{nom}}], \tag{6.2}$$

where P_j^{nom} indicates the nominal leak-free pressure at the j-th node, which is estimated by simulation using a hydraulic model of the network.

In practice, it is only necessary to compute the pressure sensitivity in the nodes with sensors because only the columns corresponding to the sensed nodes are retained. Similarly, only the residual components of the sensed nodes are calculated. On the other hand, given the difficulty of analytically calculating the partial derivatives in (6.1), the sensitivities are estimated numerically by increasing the leak flow rate from 0 to a given value Q^ℓ and measuring (by simulation) the corresponding increase in node pressures:

$$s_{ij} = \frac{\partial P_j}{\partial q_i^\ell} \approx \frac{\Delta P_j}{Q^\ell}. \tag{6.3}$$

One way to use the sensitivity matrix is to binarize it by setting a sensitivity threshold so that each sensitivity vector becomes a string of ones and zeros (Pérez et al., 2009, 2011). These binary strings constitute *leak signatures* and will be compared with the residual vector for matching. The residual vector is binarized in a similar way to the sensitivity matrix (6.1). If there is no exact match between the binary signature of a node and the binarized residual, the node with signature that has the highest number of bits matching the binarized residual is taken as the estimated position of the leak.

[1] Some authors use the transpose of the sensitivity matrix defined in (6.1) and the residual vector as a column vector.

Another proposal for locating leaks is through the statistical correlation of the actual residual with the vectors of the sensitivity matrix (Pérez et al., 2014):

$$\text{corr}(\mathbf{r}, \mathbf{s}_i) = \frac{(\mathbf{r} - \overline{\mathbf{r}})(\mathbf{s}_i - \overline{\mathbf{s}_i})^\mathsf{T}}{\sqrt{(\mathbf{r} - \overline{\mathbf{r}})(\mathbf{r} - \overline{\mathbf{r}})^\mathsf{T}}\sqrt{(\mathbf{s}_i - \overline{\mathbf{s}_i})(\mathbf{s}_i - \overline{\mathbf{s}_i})^\mathsf{T}}}, \tag{6.4}$$

where the overline $^-$ indicates the arithmetic means of the vector components. In this method, the sensitivity matrix is not binarized. Equation (6.4) aims to measure the similarity between the residual vector and the sensitivity vector of each leak node. The leak is assumed to be located at the node with sensitivity vector that has the highest correlation (closest to 1) with the residual vector. Framing it in machine learning methods, the proposal of Pérez et al. (2014) is about a 1-NN classifier with correlation distance.

A variant of the previous method was presented by Casillas et al. (2014), where the similarity is not measured with a correlation but through the angular closeness between the residual vector and the sensitivity vectors. The angle between $\mathbf{r}$ and $\mathbf{s}_i$ is given by

$$\alpha_i = \arccos\left(\frac{\mathbf{r} \cdot \mathbf{s}_i}{\|\mathbf{r}\| \, \|\mathbf{s}_i\|}\right), \tag{6.5}$$

where $\cdot$ denotes the inner product between vectors. The authors also proposed repeatedly calculating the angles along a time horizon (e.g., one day) to obtain an average angle. Finally, the leak estimated position is the node whose sensitivity vector has the smallest average angle. Framing it in machine learning methods, the proposal of Casillas et al. (2014) is about a 1-NN classifier with cosine distance.

An improvement in leak localization based on pressure residuals was presented by Casillas et al. (2015). They propose the residual vector projection towards a space of lower dimension (called *leak signature space*) in which the projections are less dependent on the leak magnitude and depend mainly on its location. The reduction in the residual space dimensionality is achieved by forcing one component of the residual vector to always be unitary. To locate the leak given a pressure residual, look for the smallest Euclidean distance between the actual projected residual and the leak signatures in the new space. Each leak signature is obtained by averaging the corresponding node's leak signatures. In terms of machine learning methods, the proposal of Casillas et al. (2015) is about an Euclidean 1-NN classifier with transformed residuals as features.

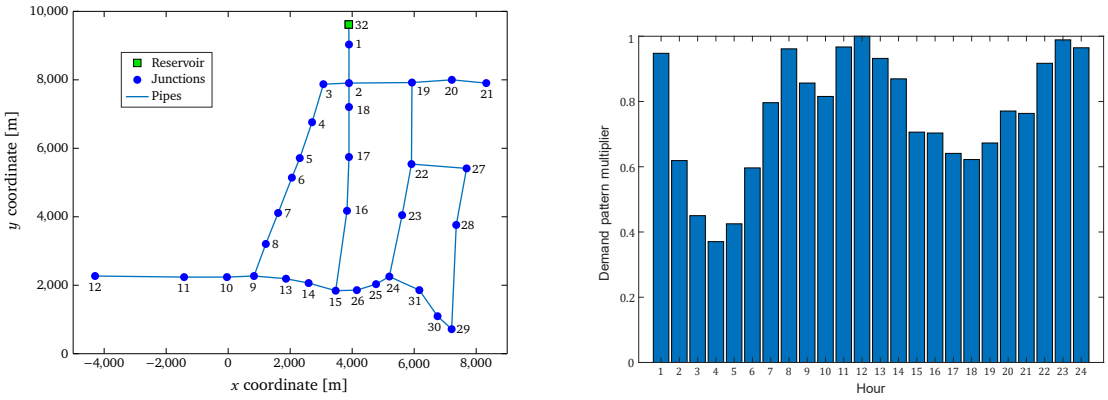
The correlation (6.4) and the angle between residuals (6.5) are the basis for discussing leak localization methods using classifiers in this chapter. Before addressing the problem of locating leaks using classifiers, the metrics used to evaluate the performance of different classifiers (or classifiers with different parameterization) will be presented.

6.3 Case studies

This section describes two water distribution networks used as case studies. The first case is a simple Hanoi network model, a network with few nodes widely used as a benchmark to evaluate various leak localization methods reported in the literature. Because of its small size, leak localization techniques can be comprehensively tested on this network. The second case is a Madrid network sector, a district

metered area (DMA), where leakage flow, node pressures, and inlet flow measurements were performed to test leak localization algorithms in real operating environments.

The Hanoi network model is composed of one reservoir, 31 junction nodes, and 34 pipes with a total length of 38.61 km organized in 3 loops, as shown in Fig. 6.1(a). No pumping facilities are considered since only a single fixed-head source (node number 32) at an elevation of 100 m is available. The Hanoi system was first presented by Fujiwara and Khang (1990) and is based on the planned trunk network of Hanoi, Vietnam. The data sets generated by simulation for this work consider leaks at all junction nodes with flow rates from 1 to 80 l/s, in addition to the leak-free nominal conditions.



(a) Geographical layout (b) Flow profile

Fig. 6.1: The Hanoi network

Because the Hanoi network model as published by Fujiwara and Khang (1990) does not include demand patterns of user consumption, a demand pattern of the Madrid DMA (described below) was used to modulate the base demand. At each time, the demand at the consumption nodes adjusts proportionally to the increase in the inflow, with variation throughout the day that is expressed by a multiplier coefficient. The flow profile used for the Hanoi network is shown in Fig. 6.1(b).

Synthetic data were obtained from the Hanoi network model using EPANET (Rossman et al., 2020; Eliades et al., 2016), which are node pressures of different leakage scenarios and at different times. The pressure data set contains a hypermatrix $\mathbf{P} \in \mathbb{R}^{24\times31\times1550}$ that captures the time variation of the node over the 24 hours of a day. One column is used for each node pressure, as shown in Fig. 6.2. The third dimension (depth) is used for the different leak scenarios. Each leak scenario refers to a combination of leak node and leak size. There are 1550 leak scenarios because 31 different leak nodes are combined with 50 different leak sizes (from 1 to 50 l/s). The data set also includes a vector $\mathbf{y} \in \mathbb{R}^{1550}$ containing the leak location for each scenario to be used in the classifier training. Additionally, the data set contains a pressured matrix $\mathbf{P}^{(0)} \in \mathbb{R}^{24\times31}$ with the information of the time variations of the node pressures under nominal leak-free conditions. The matrix $\mathbf{P}^{(0)}$ must be subtracted from each plane in $\mathbf{P}$ to obtain the pressure residuals used as features in leak classification.

The data obtained from simulations with the Hanoi network model are ideal since they consider a deterministic node demand without measurement noise. Therefore, Gaussian white noise is added to the simulated measurements under different levels. The added noise is specified by the signal-to-noise ratio

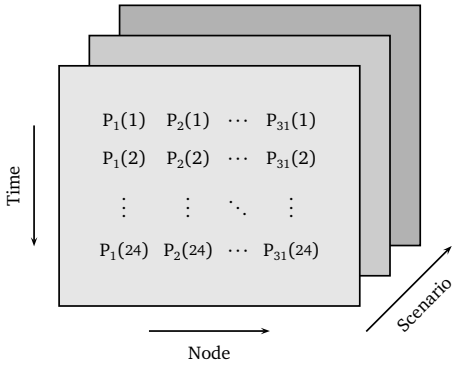
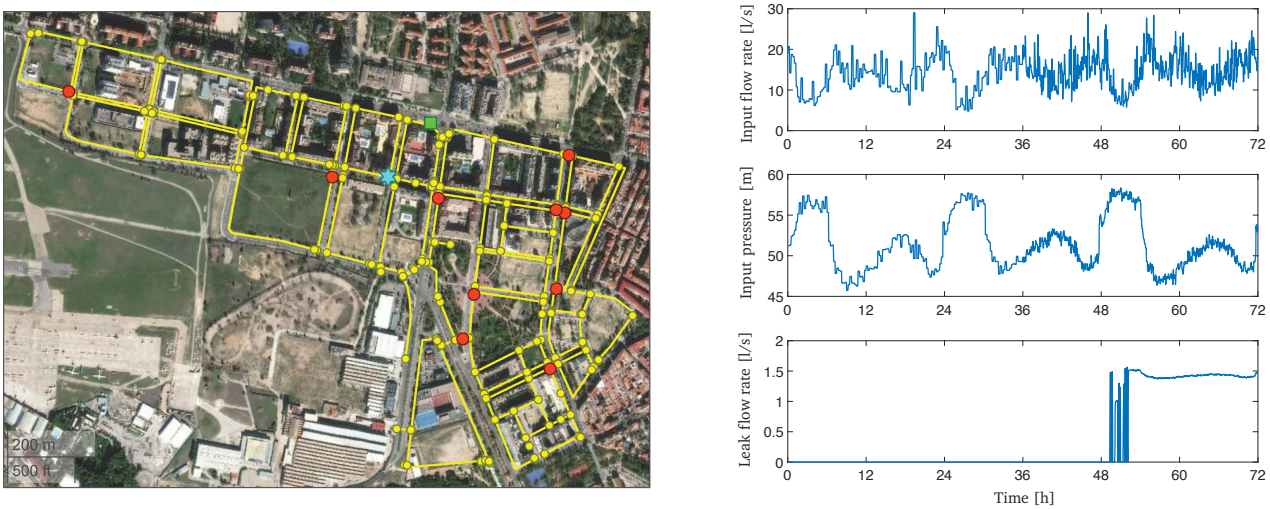


Fig. 6.2: Arrangement of the pressure hyper-matrix in the Hanoi data set

(SNR), which represents the size of the magnitude of the valid signal relative to the magnitude of the measurement noise, as defined by Box (1988):

$$\text{SNR dB} = 20 \log_{10}(\text{signal}/\text{noise}) . \tag{6.6}$$

The second case study is a DMA in the Madrid network formed by one reservoir and 312 junction nodes connected by 14224 m of pipe with diameters between 80 mm and 350 mm. The layout of this network is shown in Fig. 6.3, where the green square represents the reservoir, the 10 red circles represent installed pressure sensors, and the blue star indicates the leak node used in the tests. The mean pressure head is about 55 m, and the maximum pressure head available at the reservoir is about 58 m. Nevertheless, it varies throughout the day and is monitored by a sensor.



(a) Geographical distribution (b) Measurements for three days

Fig. 6.3: DMA in the Madrid network

Figure 6.3(b) shows the inlet flow and supply pressure of the network monitored during three days of operation, including a leak on the third day. The leak event is simulated by opening a fire hydrant at node

number 272. Figure 6.3(b) shows that generally, the supply pressure decreases when the consumption in the network increases.

The demand profile used to model user consumption on the Hanoi network was obtained from this DMA by averaging the network inflow over 10 consecutive days.

6.4 Performance metrics in leak localization

This section describes three indicators used in this work to quantify the performance of leak localization systems: confusion matrix, accuracy, and topological distance.

6.4.1 Confusion matrix

The confusion matrix is an instrument used in classification and pattern recognition to evaluate a classifier's ability to distinguish different classes or conditions of a system. This is why it is frequently used in FDI (fault detection and isolation) to assess a system/algorithm's ability to detect and isolate faults. In a classification problem with N classes, the confusion matrix is a $N \times N$ square integer matrix $\mathbf{C} = [c_{ij}]$, where c_{ij} contains the number of samples from the i-th class that are misclassified into the j-th class.

In leak localization, if each leak is associated with the closest node in the network, the rows of $\mathbf{C}$ represent the "true node" where the leak occurs (the node closest to the leak), while the columns of $\mathbf{C}$ represent the "predicted node" where leaks are located. Usually, it is represented by y the true leak node and by $\widehat{y}$ the predicted leak node (Fig. 6.4). Since the correct node (row) and the predicted node (column) should match in each test, ideally the confusion matrix should be diagonal. Nonzero items off the diagonal represent misclassifications.

Predicted leak node, $\widehat{y}$

True leak node, y	1	$\cdots$	j	$\cdots$	N
1	$c_{1,1}$		$c_{1,j}$		$c_{1,N}$
$\vdots$		$\ddots$			$\vdots$
i	$c_{i,1}$		$c_{i,j}$		$c_{i,N}$
$\vdots$				$\ddots$	$\vdots$
N	$c_{N,1}$		$c_{N,j}$		$c_{N,N}$

Fig. 6.4: Confusion matrix for leak localization

6.4.2 Classification accuracy

Although the confusion matrix provides a comprehensive summary of the classifier's ability to discriminate between leaks at different nodes, it is sometimes more convenient to summarize the overall performance in a single measure without detailing the separation between different leak classes. For this, the *classification accuracy* can be used, which is defined as the fraction of correctly classified samples in the classifier testing. In terms of the confusion matrix $\mathbf{C} = [c_{ij}]$, the accuracy (Acc) is obtained by dividing the sum of the elements on the main diagonal (the trace of $\mathbf{C}$) by the sum of all the elements:

$$\text{Acc} = \frac{\sum_{i=1}^{N} c_{ii}}{\sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij}}. \tag{6.7}$$

A value $\text{Acc} = 1$ indicates a perfect classifier performance, although in practice the accuracy is only close to 1 in the best cases. The proximity to 1, however, is only a good performance indicator when the test data set is balanced in the number of samples for each class. In leak localization tests, where classifiers are trained and tested on synthetic data, the simulations are programmed to equal the number of samples for each class (leak node).

In order to optimize leak localization, the performance of localization algorithms is often expressed in terms of *classification error* (classification loss), which is complementary to accuracy:

$$\text{Loss} = 1 - \text{Acc}. \tag{6.8}$$

A cross-validation strategy is also used to assess leak localization using classifiers in this work. The cross-validation procedure consists of partitioning the training data set into K groups and iterating K times, taking one of them as a test set and the rest as the training set. At each iteration, a classification loss is calculated; at the end, all losses are averaged to obtain a robust measure of performance called K-fold loss.

A limitation of the accuracy and the classification loss as performance metrics in leak localization is that they only consider the number of errors when locating leaks but not the magnitude of these errors. The following metric is intended to evaluate the performance of leak localization methods in a more comprehensive way.

6.4.3 Topological distance

The topological distance between two nodes is the number of links in the shortest path connecting them. Formally, a hydraulic network is a graph composed of a set of nodes $\mathscr{N} = \{N_k\}$ and a set of links $\mathscr{L} = \{L_k\}$, where each link (pipe) connects a pair of nodes. For any two nodes, i and j, the topological distance between them is denoted by d_{ij}. If the shortest path between this pair of nodes is represented by the link subset $\mathscr{P}_{ij} \subset \mathscr{L}$, then

$$d_{ij} = \left| \mathscr{P}_{ij} \right| \tag{6.9}$$

where $|\cdot|$ indicates cardinality.

In Definition (6.9), all links in the shortest path are equally valued; no link is considered more significant than another. It is also possible to define a length-weighted topological distance where the length of the pipe in each link of the shortest path is considered:

$$d_{ij} = \sum_{L_k \in \mathscr{P}_{ij}} \text{length}(L_k), \tag{6.10}$$

where $\text{length}(L_k)$ is the length (m) of the k-th link. Considering all the different pairs of nodes in a network, the topological distances are organized in a matrix. Since $d_{ij} = d_{ji}$, $\mathbf{D} = [d_{ij}]$ is a symmetric matrix. Furthermore, it has a null diagonal since $d_{ii} = 0$.

The topological distance between nodes is a metric for the leak localization error, using d_{ij} to quantify the error when a leak at node i is located at node j. It is also used as a cost function for hyperparameter optimization in training a classifier for leak localization. A series of data is frequently tested considering different leak scenarios, varying the leak location and magnitude. For each entry in the data set, a predicted leak node, $\widehat{y}$, is obtained and compared against the real leak node, y. A metric that summarizes the overall performance of the localization method on the test data is the *average topological distance* (ATD)

$$\text{ATD} = \frac{\sum_i \sum_j c_{ij}\, d_{ij}}{\sum_i \sum_j c_{ij}}, \tag{6.11}$$

where $[c_{ij}]$ is the confusion matrix described in Subsection 6.4.1.

Ideally, in the best case of leak localization, both ATD and classification loss should be 0, whereas the accuracy should tend to 1.

The topological distance between a pair of nodes can be obtained, among other methods, by Bread-First computation (Akiba et al., 2013) when all the links are treated with unit weight or by the Dijkstra algorithm (Dijkstra, 1959) when the pipe lengths are used to weigh the links.

6.5 Classification method based on Bayesian decision

Leak localization is presented in this work as a multiclass classification problem using supervised learning. The proposed classifiers can be framed within Bayes' decision theory and computed using probabilities. In this framework, the creation and tuning consist of proper probability density functions from training data. The Bayesian decision process is summarized below.

Let us consider a classification problem with J classes, determined by feature vectors $\mathbf{x} = [x_1, \ldots, x_n]$ representing observations of n features. The classifier response is a class label denoted by y, so y_j means that a given data sample belongs to the j-th class. To classify an observation $\mathbf{x}$, the *posterior probability* $P(y_j \,|\, \mathbf{x})$ must be estimated, which is the probability that the given observation belongs to the class y_j. It makes sense to assign an observation to the class with the highest posterior probability. Therefore, a way to estimate or fit these probabilities is required. Using the Bayes' theorem, the posterior probability can be computed as follows:

$$P(y_j \,|\, \mathbf{x}) = \frac{P(y_j)\, P(\mathbf{x} \,|\, y_j)}{P(\mathbf{x})}, \tag{6.12}$$

where

$$P(\mathbf{x}) = \sum_{j=1}^{J} P(y_j)\, P(\mathbf{x} \,|\, y_j) \tag{6.13}$$

is a common denominator for all classes and is called *evidence*. The factors $P(y_j)$ and $P(\mathbf{x} \,|\, y_j)$ in the above equations are the *prior probability* and *class-conditional probability* of class y_j, respectively. To distinguish between the concepts of prior and posterior probabilities, the rest of this section will use $P_0(\cdot)$ for prior and $\widehat{P}(\cdot)$ for posterior.

When there is no prior knowledge about the frequency of occurrence of each class, the prior probabilities $P_0(y_j)$ of each class can be assigned **uniformly** assuming that the cardinality of the classes is J and all members are equally probable:

$$P_0(y_j) = \frac{1}{\text{number of classes}} = \frac{1}{J}, \tag{6.14}$$

or it can be calculated **empirically** from the relative frequency of the class in the training set

$$P_0(y_j) = \frac{\text{number of samples in the } j\text{-th class}}{\text{total number of samples}}. \tag{6.15}$$

Once the prior probabilities and the class-conditional probabilities have been calculated, the predicted class is determined by the rule below.

Bayes decision rule

Given a feature vector $\mathbf{x}$, assign it to class y_j if

$$\widehat{P}(y_j \,|\, \mathbf{x}) > \widehat{P}(y_i \,|\, \mathbf{x}); \quad i = 1, 2, \ldots, J; i \neq j. \tag{6.16}$$

This means that the predicted class, denoted by $\widehat{y}(\mathbf{x})$, is given by

$$\widehat{y}(\mathbf{x}) = \arg\max_{j} \widehat{P}(y_j \,|\, \mathbf{x}), \tag{6.17}$$

which is also called the MAP (*maximum a posteriori*) decision rule.

In a diagnosis application, not all misclassifications have the same impact in the context of the problem. If some classification errors need to be weighted more than others, then the classification output can be computed by minimizing the *expected classification cost*,

$$\widehat{y}(\mathbf{x}) = \arg\min_{i=1,\ldots,J} \sum_{j=1}^{J} \widehat{P}(y_j \,|\, \mathbf{x})\, \Gamma_{ij}, \tag{6.18}$$

where Γ_{ij} is the cost of classifying an observation as class y_i when its true class is y_j. The simplest way to assign the cost of misclassification is

$$\Gamma_{ij} = \begin{cases} 0 & \text{if } i = j, \\ 1 & \text{if } i \neq j, \end{cases} \tag{6.19}$$

where all misclassifications are penalized equally; however, sometimes it is reasonable to use penalties other than 1. In applications for locating water leaks, where the classes are associated with the leak positions within the network, it is reasonable to increase the misclassification cost as the distance between the real leak position and the position predicted by the classifier.

All classification techniques used in this work can be framed within Bayes' decision theory, although they differ in estimating the class conditional probabilities. These techniques are naive Bayes classification, discriminant analysis, k-nearest neighbors (k-NN), and decision trees.

Naive Bayes classification and discriminant analysis compute the likelihood using Gaussian distributions. Classification by discriminant analysis assumes multivariate normal distributions for class-conditional probability densities, so the data are modeled using a Gaussian mixture distribution. Conversely, naive Bayes classification assumes that each feature x_i is independent of the other features, so the joint probability $P(\mathbf{x} \mid y_j)$ can be expressed from the product of the conditional probabilities of each feature individually.

In naive Bayesian classification and discriminant analysis, the distribution parameters (mean and variance/covariance) are estimated from the training data, so they are parametric classification methods. Furthermore, the classification by decision trees and k-NN is based on non-parametric models. The similarities/differences and the implementation details of each of these methods can be found in Martinez and Martinez (2015).

The Bayesian decision approach described in this section will be used both to estimate the leak location at a specific time point (Sections 6.7 and 6.8) and also to propagate the probability of leak occurrence between different time points (Section 6.9). The features x_i are obtained from the pressures in the nodes equipped with sensors, and the class labels y_j correspond to the leaky nodes.

Before describing the leak localization using classifiers, methods based on the sensitivity matrix are introduced since these have motivated the proposal.

6.6 Leak localization using the sensitivity matrix

Leaks in a network are manifested as pressure losses in the nodes. Therefore, the differences between the pressures before and after a leak event, the residuals, provide information for diagnosing the event. Leaks in the network, however, do not affect all nodes equally since the proportion in which the pressure residual of each node varies depends on the magnitude of the residual and its location.

From now on, matrices will be used to relate the leak position with the position of the pressure sensor where the residuals are computed. The leak node and the sensor node will be called i (row) and j (column), respectively. The matrix $[s_{ij}] = \partial P_j / \partial q_i$, as defined in (6.1), summarizes the pressure sensitivity at the j-th node to leakage at the i-th node. The heat map in Fig. 6.5 shows the sensitivity of all node pressures to leakage at each node in the Hanoi network using colors from white (lower sensitivity) to black (higher sensitivity), passing through yellow and red hues. It can be seen that nodes 21, 27, 28, 29, and 12 have high sensitivities, whereas node 1 has very low sensitivity. Furthermore, each node's pressure is more sensitive to leaks located at the same node. Therefore, the values on the matrix diagonal are dominant.

The sensitivity matrix, calculated by simulation, characterizes the influence of each possible leak on different node pressures. Thus, when a leak has already been detected, its location can be estimated by comparing the actual pressure residuals' similarity to the sensitivity vectors for each potential leaky node.

In order to measure the similarity between the residual vector and the sensitivity vectors, their Euclidean distance, their correlation, and the angle between them can be used. The Euclidean distance depends greatly on the magnitude of the leak and little on its location, so its use for leak localization requires that the residuals be mapped to a vector space where the magnitude dependence is minimized.

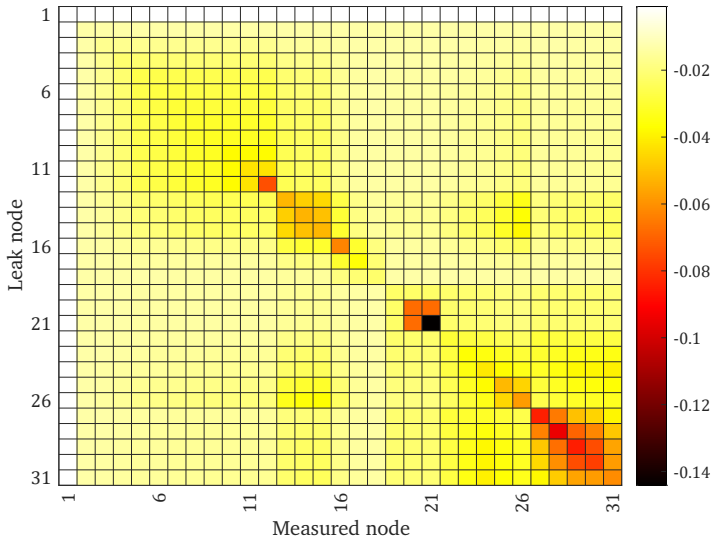


Fig. 6.5: Sensitivity matrix of Hanoi network

A similarity measure (between residuals and sensitivity vectors) little affected by the leak's magnitude is correlated. With this measure, the predicted leak node is the one with sensitivity vector $\mathbf{s}_i$ that has the highest correlation with the actual residual $\mathbf{r}$:

$$\widehat{y}(\mathbf{r}) = \underset{i\in 1,2,\ldots,n}{\arg\max}\ \mathrm{corr}(\mathbf{s}_i, \mathbf{r}). \tag{6.20}$$

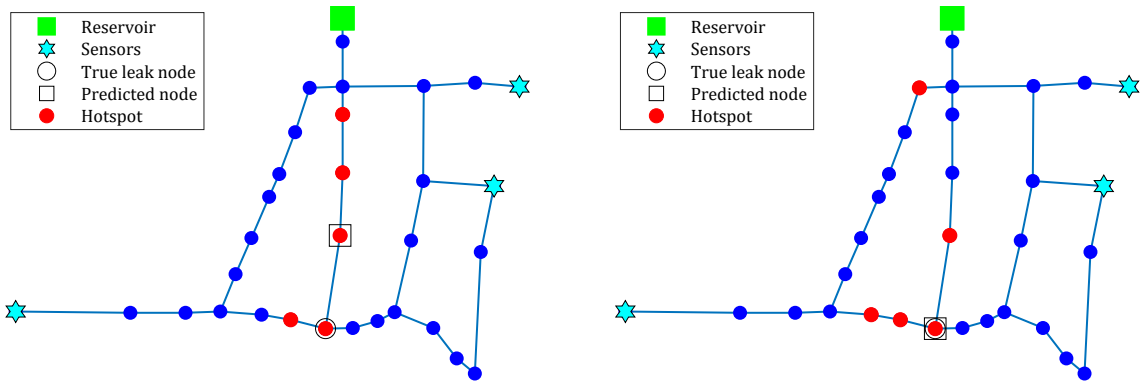
Since there is uncertainty in the leak node prediction, a list containing K nodes ($K > 1$) with the highest correlations in (6.20) can be considered to mark a "hotspot" of the leak in the network.

Another way to estimate the leak location is by computing the angle between the residual vector $\mathbf{r}$ and each of the sensitivity vectors $\mathbf{s}_i$. The node i with the minimum angle between $\mathbf{s}_i$ and $\mathbf{r}$ is assumed as a leak node. In practice, it is not necessary to compute the angle α_i between $\mathbf{s}_i$ and $\mathbf{r}$ but only the cosine of this angle. The node for which the cosine is maximum is assumed as the leak node (an angle is a minimum if its cosine is maximum):

$$\widehat{y}(\mathbf{r}) = \underset{i\in 1,2,\ldots,n}{\arg\min}\ \alpha_i = \underset{i\in 1,2,\ldots,n}{\arg\max}\ \cos(\alpha_i) = \underset{i\in 1,2,\ldots,n}{\arg\max}\ \frac{\mathbf{r}\cdot\mathbf{s}_i}{\|\mathbf{r}\|\,\|\mathbf{s}_i\|}. \tag{6.21}$$

A list containing K nodes ($K > 1$) with the highest cosines in (6.21) are considered to mark a hotspot for the tested leak.

Figure 6.6 shows the results of locating a leak in the Hanoi network using both methods: maximum correlation and minimum angle. As can be seen, the maximum correlation method does not locate the exact node of the leak but estimates a consistent hotspot. In contrast, the minimum angle method locates the exact leak node but estimates a more dispersed hotspot. A sensitivity matrix with $\Delta q = 25$ l/s was used. For the localization test, a leakage of 18 l/s was simulated at node 15 and noise of SNR = 60 dB added to the pressure sensors at nodes 12, 21, and 27.



(a) Prediction by maximum correlation (b) Prediction by minimum angle

Fig. 6.6: Leak locations obtained by maximum correlation and minimum angle in the Hanoi network

The maximum correlation and minimum angle methods have a common characteristic: they find the training sample with the most significant similarity (the minimum correlation/cosine distance). A leak localization method using k-NN classifiers is considered in the next section to generalize this approach to other distance metrics.

6.7 Leak localization using k-NN with hyper-parameter optimization

The k-NN (k-Nearest Neighbors) method classifies test data by taking the most frequent class among the k feature vectors most similar to the test data within the training set. In the simplest form of k-NN, where $k = 1$, only the sample from the training set that is closest to the test sample (nearest neighbor in feature space) is used. The test sample is then assumed to belong to the same class as the nearest training sample. Different distance metrics can measure the similarity between two feature vectors, including the well-known Euclidean distance.

The measure used to quantify the nearness between points significantly influences the k-NN classifier's performance. Although Euclidean distance is the most common way to measure dissimilarity between data, it is also possible to use other dissimilarity measures that express other types of distance between points in feature space. Some frequently used distance metrics are listed in Table 6.1, where $\mathbf{x}$ and $\mathbf{x}'$ are assumed to be feature vectors in $\mathbb{R}^n$.

Table 6.1: Distance metrics in k-NN classification

Distance name	**Definition**
Euclidean distance	$d(\mathbf{x},\mathbf{x}') = \sqrt{(\mathbf{x}-\mathbf{x}')^{\mathsf{T}}(\mathbf{x}-\mathbf{x}')}$
Mahalanobis distance	$d(\mathbf{x},\mathbf{x}') = \sqrt{(\mathbf{x}-\mathbf{x}')^{\mathsf{T}}\mathbf{S}^{-1}(\mathbf{x}-\mathbf{x}')}$
Manhattan (city block) distance	$d(\mathbf{x},\mathbf{x}') = \sum_{i=1}^{n}\lvert x_i - x_i'\rvert$
Chebyshev distance	$d(\mathbf{x},\mathbf{x}') = \max_i \lvert x_i - x_i'\rvert$
Minkowski distance	$d(\mathbf{x},\mathbf{x}') = (\sum_{i=1}^{n}\lvert x_i - x_i'\rvert^p)^{1/p}$
Correlation distance	$d(\mathbf{x},\mathbf{x}') = 1 - \mathrm{corr}(\mathbf{x},\mathbf{x}')$
Cosine distance	$d(\mathbf{x},\mathbf{x}') = 1 - \cos(\mathbf{x},\mathbf{x}')$

The Manhattan, Euclidean, and Chebyshev distances in Table 6.1 are special cases of the Minkowski distance for values $p = 1$, $p = 2$, and $p = \infty$, respectively. The matrix $\mathbf{S}$ on the Mahalanobis distance is the data covariance (De Maesschalck et al., 2000). On the other hand, $\mathrm{corr}(\mathbf{x},\mathbf{x}')$ is the Pearson's linear correlation between $\mathbf{x}$ and $\mathbf{x}'$, treated as sequences of values, and $\cos(\mathbf{x},\mathbf{x}')$ denotes the cosine of the angle between the $\mathbf{x}$ and $\mathbf{x}'$ vectors in the n-dimensional feature space, as defined in (6.5). A cosine value of 0 means that the two vectors are orthogonal and have no match. The closer the cosine value is to 1, the smaller the angle and the greater the match between vectors.

Fine-tuning a k-NN classifier should include determining the most appropriate distance metric for the problem and determining the number of neighbors (k) to consider. The distance metric and the number of neighbors are known as k-NN *hyperparameters*.

The performance of a k-NN classifier depends on the number of nearest neighbors to use. Small values of k tend to produce an overfitted classifier that is very sensitive to measurement noise in features. In general, k-NN considers a higher number of neighbors, $k > 1$, which allows a more robust classification (less sensitive to outliers and measurement noise).

The general idea of leak localization using classifiers is to fit a predictive model that estimates the leak node location (class label) based on a set of node pressures (features) measurements. Next, leak location is explored in this section using k-NN as the first classification method. Unlike the methods described in the previous section, the k-NN classification generalizes the leaky node search by considering any number of nodes with similar leaks and for any distance metric.

From the Hanoi network pressure data set, where leaks of different flow rates are considered in each node of the network, k-NN classifiers were trained considering two different sets of features. In the first set, the pressures measured at the nodes were used directly, whereas in the second set the pressure residuals were used. For each set of features, the effect of varying the number of neighbors has been analyzed as well as the distance metric used to measure the nearness between neighbors. The results presented in this section were obtained assuming pressure sensors placed in nodes 12, 21, and 27. The procedure to train and test the k-NN classifier on the Hanoi network is described in the following box.

Training and testing k-NN on the Hanoi network

1. Build the pressure matrix $\mathbf{P} = [\mathbf{P}_{12}, \mathbf{P}_{21}, \mathbf{P}_{27}]$, containing one column for each sensor and one row for each leak scenario. In addition, create a vector $\mathbf{y}$ with class labels (leak nodes) containing one element for each row of $\mathbf{P}$. The matrix $\mathbf{R} = \mathbf{P} - \mathbf{P}_0$ must also be computed when pressure residuals are used as features.
2. Divide the data ($\mathbf{P}$, $\mathbf{R}$ and $\mathbf{y}$) into two groups; one for training and the other for testing. For training, the samples corresponding to leakage flow $q_\ell = 2, 4, \ldots, 50$ l/s are used, whereas for testing the samples corresponding to leakage flow $q_\ell = 1, 3, \ldots, 49$ l/s are used. In cross-validation tests, the entire data set is taken for training, and testing is performed on the validation subset in each iteration.
3. Train the k-NN classifier using $\mathbf{P}_{\text{train}}$ or $\mathbf{R}_{\text{train}}$, depending on the set of features to be used. A distance metric and a number of neighbors must be specified.
4. Test the trained classifier using the testing data $\mathbf{P}_{\text{test}}$ or $\mathbf{R}_{\text{test}}$. Predicted leak nodes are stored in a vector $\widehat{\mathbf{y}}$ to later evaluate the classifier performance.
5. Compute the classification loss (error) and the confusion matrix.

In order to find the best hyperparameters (the best combination of distance and number of neighbors) of the k-NN classifier, the classification error, as defined in (6.8), was calculated for each combination of distance and number of neighbors from $k = 1$ to $k = 50$. For each combination, a 5-fold cross-validation was applied so that the K-fold loss is used as an error metric. The K-fold loss is obtained by averaging the losses obtained on each fold. The K-fold loss for each combination of distance and number of neighbors is shown in Fig. 6.7. The definition of each distance metric was presented in Section 6.5.

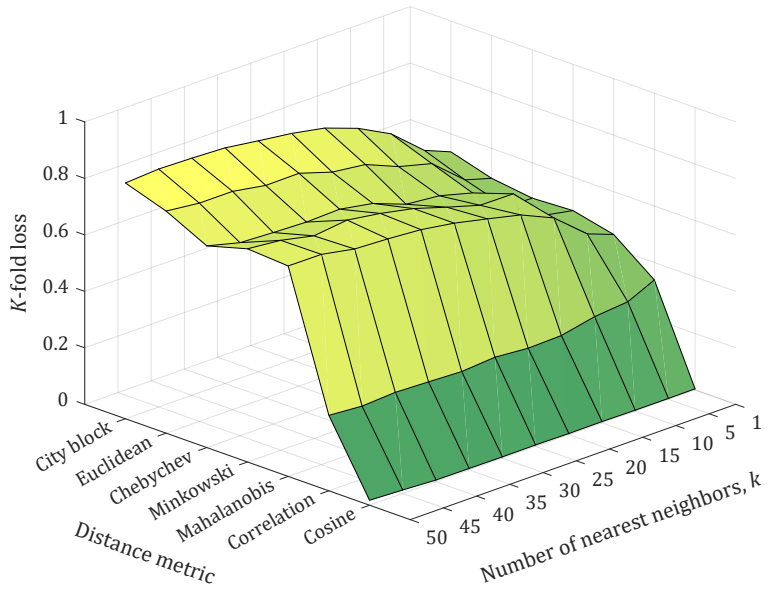


Fig. 6.7: Hyperparameter optimization in k-NN classifier for leak localization

Figure 6.7 shows that the best performance (the lowest classification error) is obtained using the cosine distance, followed by the correlation distance. Regarding the number of neighbors, the slightest classification error is obtained for 4 and 5 neighbors ($k = 4$ was selected for simplicity). Because the variation in the classification error for a different number of neighbors is barely perceptible in Fig. 6.7 for the cosine distance, this variation has been plotted separately in Fig. 6.8. This figure shows that as the number of neighbors approaches 50, the classification error increases because the data set used only contains 50 samples for each leak class.

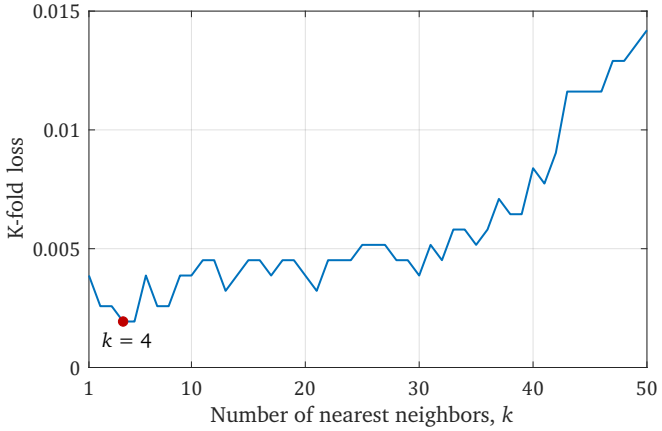


Fig. 6.8: Leak localization error using k-NN for different number of neighbors

Performance tests were applied to the k-NN classifier considering Gaussian noise in the measured pressures. The lowest classification error, obtained in noise-free conditions (SNR $= \infty$), was 0.4916 when using pressures as features, whereas it was 0.0039 when using residuals. The results plotted in Fig. 6.9 show that around 60 decibels a cut-off point is detected in the error curve where the effect of noise increases considerably. k-NN is inferior when using the measured pressures as features since almost half of the samples are wrongly classified even without noise. The use of pressure residuals (orange line) instead of raw pressures (blue line) significantly reduces leak classification error. The confusion matrices in Fig. 6.10 show an improvement in leak localization when pressure residuals are used instead of raw pressures.

From the results obtained in this section, it can be concluded that pressure residuals are an excellent option as features to classify leaks by their location. Furthermore, it was found that the best way to measure the similarity between residual pressure samples is by utilizing the cosine distance. It is also important to emphasize that when locating leaks using k-NN several neighbors greater than one must be considered according to the pressure measurements amount of noise.

6.8 Feature transformation in classifier-based leak localization

The previous section shown that the cosine distance, determined by the angle between residuals, is a good measure of similarity to classify leaks. This means that the leaks present a characteristic directionality (in the residuals space) according to the node where they occur. This section proposes a transformation on

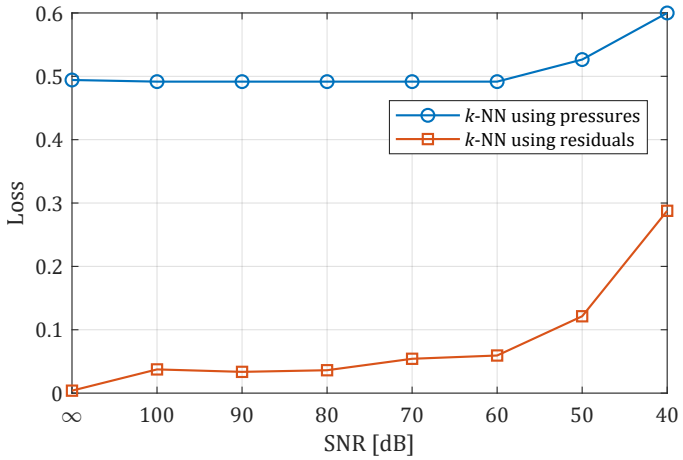


Fig. 6.9: Dependence of the classification error on the signal-to-noise ratio for k-NN with 4 nearest neighbors

the residuals to extract the information about their direction in a set of descriptors used as new features for the classifier. As shown at the end of the section, this transformation extends the concept of cosine distance used in any classifier, not just the k-NN.

According to the literature review, leak localization methods formulated as multiclass classification problems use pressure values or pressure residuals as features under a machine learning approach. An exploratory analysis of the residuals suggests that leaks at the same node tend to show a characteristic direction; see Fig. 6.11. This is a fundamental hypothesis in model-based leak localization methods based on sensitivity matrices (Pérez et al., 2011). Although in Fig. 6.11 it appears that the residuals of all the leaks at a specific node follow a constant direction, the direction of the residual vector can vary if an extensive range of leakage magnitudes is considered.

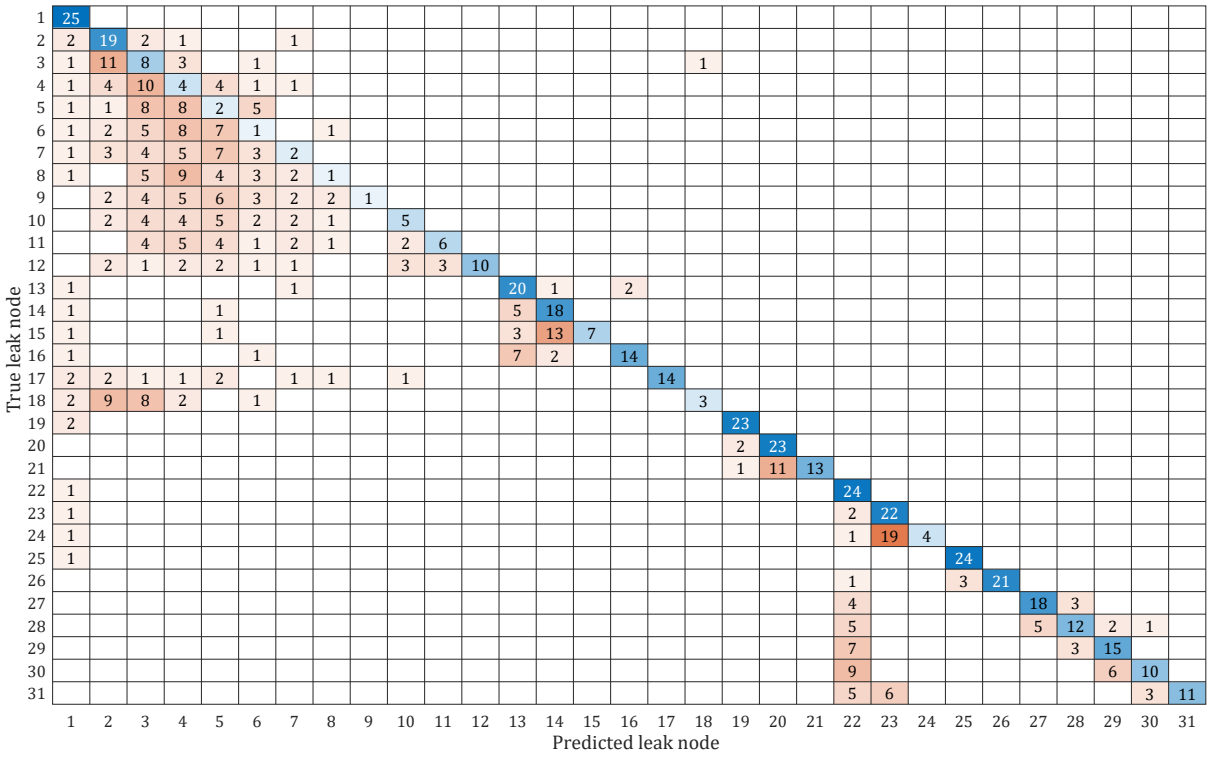
Traditional classifiers for leak localization use residual vectors in the Cartesian form where the residual magnitude and direction are implicitly combined. This does not facilitate the classification of leaks since the residual vector magnitude does not provide information on its direction. Therefore, to improve classifiers' performance for leak localization, it is proposed to map the features into a new subspace capturing only the information on the direction of the leaks and discarding the information on their magnitude.

According to vector analysis, vectors can also be expressed in a form where information about magnitude and direction is decoupled (Young, 2017). Thus, for any residual vector $\mathbf{r} = [r_1, r_2, \ldots, r_S]$, the decoupled expression is

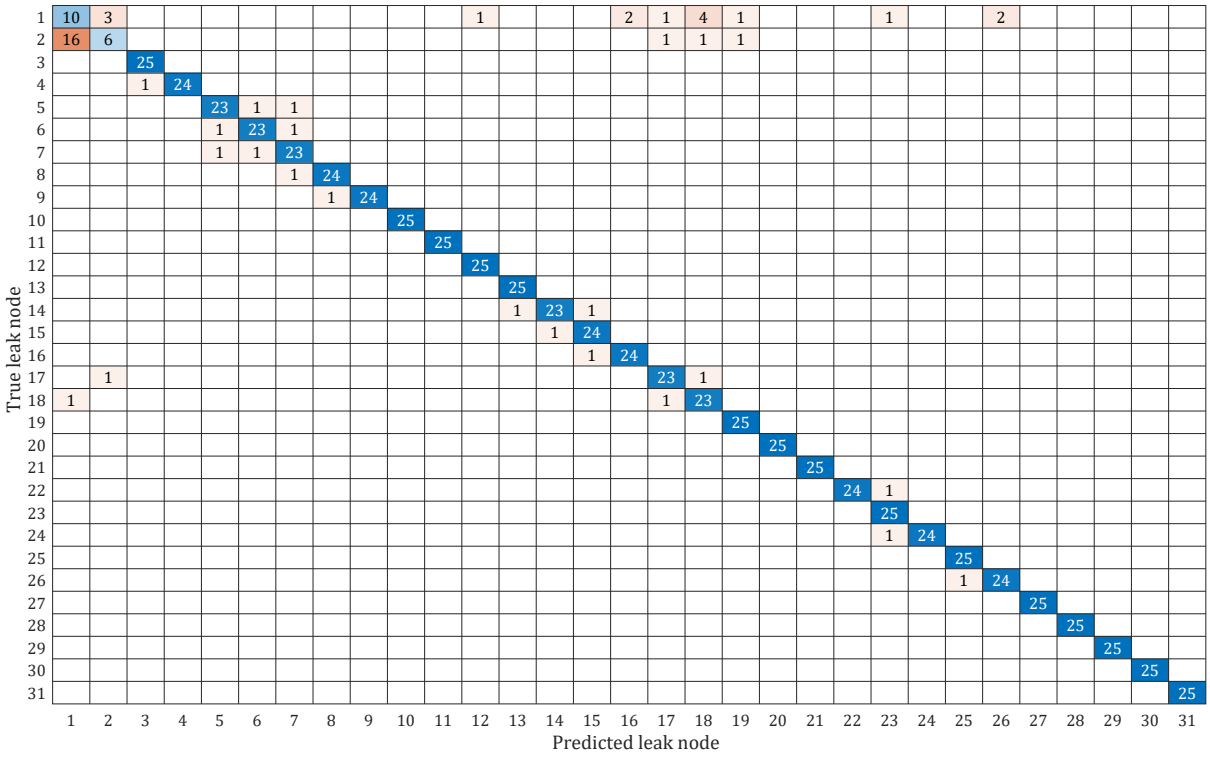
$$\mathbf{r} = M\,[\cos\theta_1, \cos\theta_2, \ldots, \cos\theta_s] = M\,[c_1, c_2, \ldots, c_S]\,, \tag{6.22}$$

where M is a scalar, and

$$c_k = f_k(r_1, r_2, \ldots, r_S) = \frac{r_k}{\sqrt{r_1^2 + r_2^2 + \cdots + r_S^2}}, \quad \text{for } k = 1, 2, \ldots, S \tag{6.23}$$



(a) Features: Pressures



(b) Features: Pressure residuals

Fig. 6.10: Confusion matrices for k-NN leak localization from noisy pressures with an SNR of 60 dB

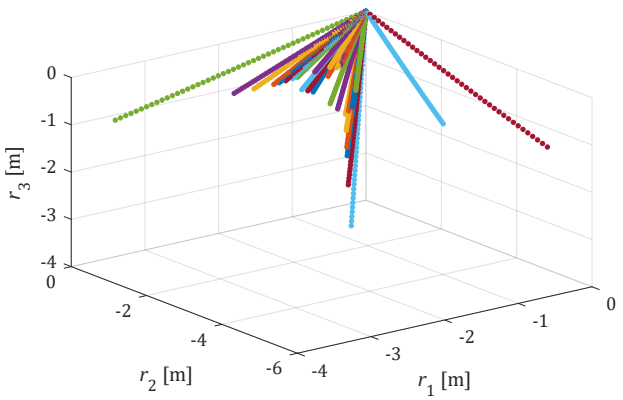


Fig. 6.11: Leaks at different nodes plotted on the residual subspace. A single color is used for each leak node, although some colors are repeated.

are the so-called **direction cosines** that uniquely describe the direction of the residual vector $\mathbf{r}$ in the S-dimensional subspace. The vector $\mathbf{c} = [c_1, c_2, \ldots, c_S]$ can be used as a new feature vector instead of the residual vector $\mathbf{r}$.

The training procedure and the predictive use of classifiers to locate leaks have been previously described by Ferrandez-Gamot et al. (2015). The modification proposed consists of using a new set of features. As shown later, replacing the original features r_k with the new features c_k improves classifiers' performance for leak localization, increasing class separability. In the diagram in Fig. 6.12, the gray box shows where the feature transformation is applied to improve the classifier's performance in the leak localization process.

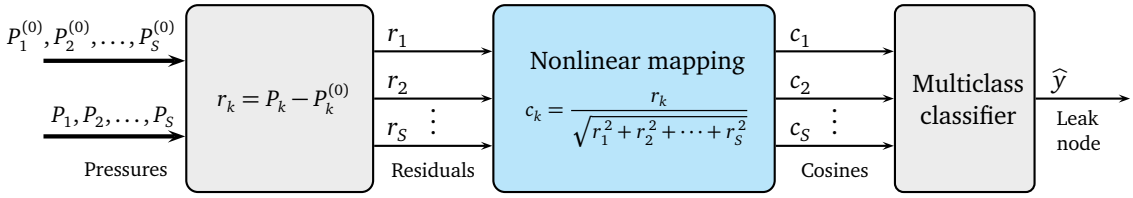


Fig. 6.12: Feature transformation to improve the classifier performance for leak localization

The proposed feature transformation can be seen as an ad hoc transformation because the original features are projected into another vector space through a nonlinear transformation. This proposal, however, does not increase the space dimension as it usually happens when the "kernel trick" (Koutroumbas and Theodoridis, 2008) is applied.

The use of direction cosines as classifiers is also related to the use of the cosine distance in k-NN. However, in k-NN, the angular proximity is measured between pairs of samples. In contrast, the direction cosines can be computed for individual samples without an additional reference sample since $[c_1, c_2, \ldots, c_S]$ measures the angular closeness of that sample concerning the implicit coordinate axes.

A series of tests verified the improvement in leak localization by using direction cosines instead of unprocessed residuals. In order to quantify the improvement obtained, the following *percentage improvement* was used

$$I\% := 100\frac{L_{\text{res}} - L_{\cos}}{L_{\text{res}}}, \tag{6.24}$$

where L_{res} is the classification loss (i.e., the leak localization error) obtained using residual features, and $L_{\cos}$ is the classification loss when using cosine features. In leak localization tests in the Hanoi network, using the data set described in Section 6.3 (sensors at nodes 12, 21, and 27), the classification loss using k-NN decreased by 99.3% when cosines were used, relative to when residuals are used directly as features. Using the naive Bayes, decision tree, and linear discriminant classifiers, the classification loss was reduced by 99.7%, 99.6%, and 97.0%, respectively, using the percentage improvement (6.24).

The lower classification loss obtained when cosines are used as features leads to better class separability. Therefore, with the four classifiers tested, the cosines better captured the directionality of the leaks in the residual subspace. This is shown in Fig. 6.13, where it is found that the class regions (where the class is detected) for three leak classes in the Hanoi network using cosine features were better defined than those for untransformed residual features. As shown in Fig. 6.13, using cosine features, the leak localization problem becomes almost independent of the classifier used since the class regions are similar for different classifiers. In fact, with cosine features, the classification loss (and consequently the accuracy) differs by less than 1% in the classifiers tested.

In order to analyze the robustness of the classifiers fed by cosine features, leak localization tests were performed considering measurement noise in node pressures. The results in Table 6.2 show that the percentage improvement when using the cosine features concerning the residual features is more significant than 50% for a wide noise margin. The percentage improvement decreases as the noise ratio in the signal increases because when the noise is considerably high (SNR < 40 decibel), the fake direction captured by the cosines become irrelevant to locate the leak. In the worst case, however, the improvement percentages remain close to 0, which means that cosine features are not decreasing in performance.

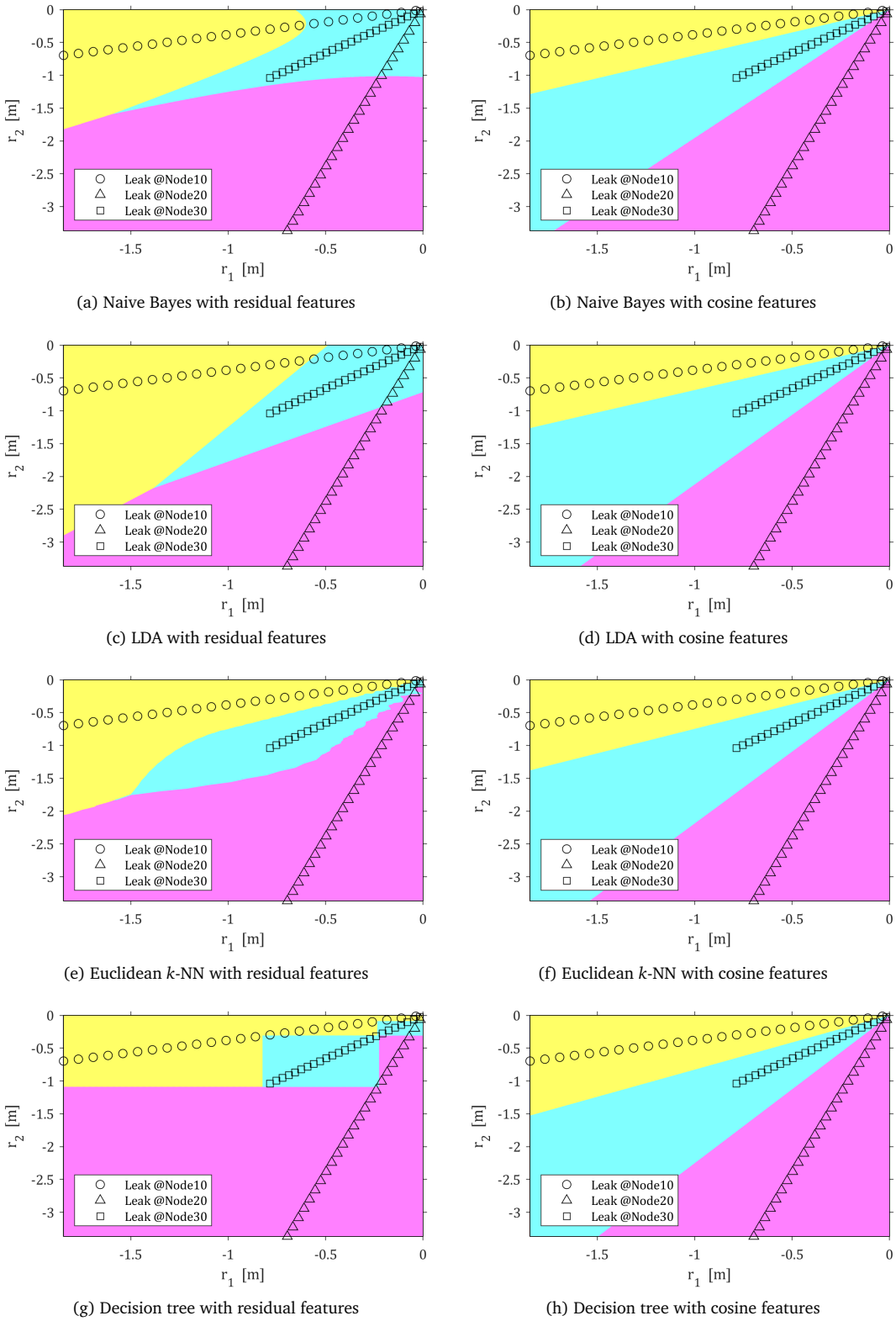
Table 6.2: Percentage improvement in performance of classifiers using cosine features

Classification method	**SNR**					
	$\infty^{\dagger}$	80 dB	60 dB	40 dB	20 dB	0 dB‡
k-Nearest Neighbors	0.993	0.873	0.394	0.006	0.013	0.005
Naive Bayes	0.997	0.939	0.405	0.009	-0.003	0.003
Decision Tree	0.996	0.917	0.473	0.021	-0.018	-0.007
Linear Discriminant Analysis	0.970	0.929	0.550	0.016	-0.040	-0.018

† Noise-free measurements

‡ Noise and signal have the same magnitude

To assess the overall performance of cosine-based classifiers in leak localization, the average topological distance (ATD) was calculated, as defined in (6.11). The results are presented on the left side of Table 6.3 for noise-free measurements, confirming the best performance when using the cosine features. By using noisy measurements, the ATD increases as the table on the right side of Table 6.3 shows, but its variation is consistent with the increase in the classification error.



(a) Naive Bayes with residual features

(b) Naive Bayes with cosine features

(c) LDA with residual features

(d) LDA with cosine features

(e) Euclidean *k*-NN with residual features

(f) Euclidean *k*-NN with cosine features

(g) Decision tree with residual features

(h) Decision tree with cosine features

Fig. 6.13: Comparison of decision boundaries for different classifiers with residual and cosine features

Table 6.3: Average topological distance using different feature sets

Noise-free measurements

Classification method	**Features**	
	Residuals	Cosines
k-Nearest Neighbors†	0.6555	0.0026
Naive Bayes	2.4090	0.0026
Decision Tree	1.3239	0.0026
Linear Discriminant Analysis	2.1974	0.0232

† Euclidean distance, $k = 5$

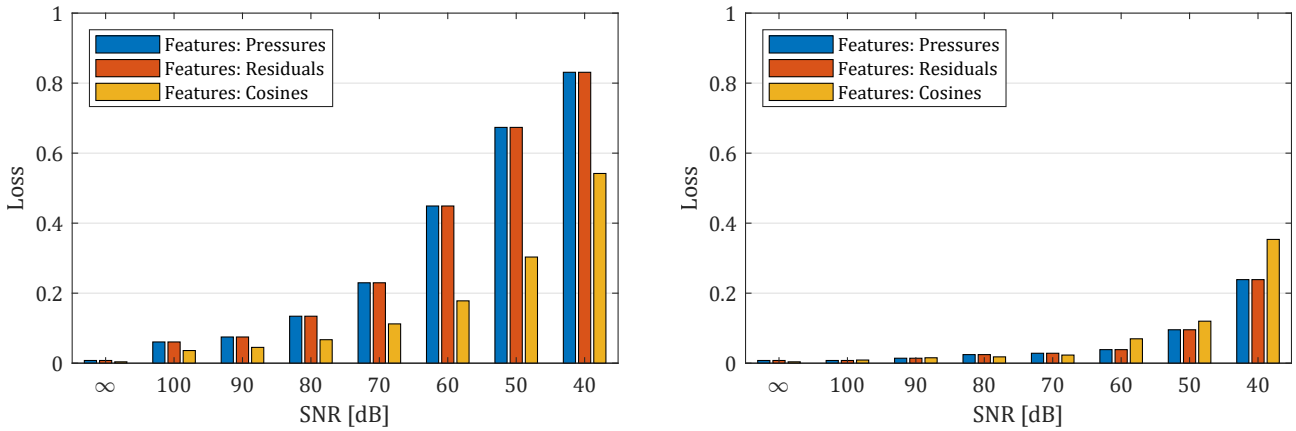
Noisy measurements, SNR = 60 dB

Classification method	**Features**	
	Residuals	Cosines
k-Nearest Neighbors†	0.9523	0.5948
Naïve Bayes	2.4387	0.9742
Decision Tree	1.3935	0.7303
Linear Discriminant Analysis	2.2090	0.7432

† Euclidean distance, $k = 5$

The quadratic discriminant analysis (QDA) classifier (not included in previous tables) deserves a separate mention in the comparative analysis of classifiers. Among the classifiers tested, this is the only one that produces good results in locating leaks using pressures without computing residuals. When residuals or cosines derived from them, however, are not used, this classifier is more sensitive to the operating point (user consumption, time of day). Therefore, its use in leak localization may require several classifiers fitted with different training sets as the operating point changes. This implies a bank of classifiers, a classifier for each hour of the day or each user consumption range.

Figure 6.14 shows the error in leak localization obtained with this classifier for different signal-to-noise ratios in the measured pressures, considering a specific time of day. This classifier's characteristic is that it showed better performance when trained from pressure measurements with added noise; see right side of Fig. 6.14. This can be explained because when the QDA classifier is trained with clean data, the class probability densities are fitted to a tiny region.



(a) Trained with noise-free data

(b) Trained with noisy data

Fig. 6.14: Classification loss in QDA-based leak localization

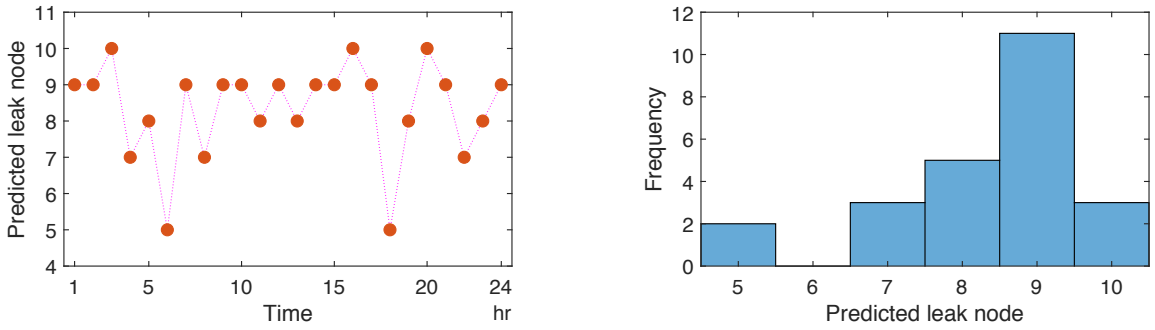
6.9 Leak localization over a time interval

Since a leak is not immediately located when it appears, a time interval T with measurements from pressure sensors is available. In this case, the features and responses are available as a time series. A refined output at a specific sample time t can be obtained, considering the class predicted during the previous interval $[t-T,\ldots,t-1,t]$. One way to refine the output is to train different classifiers for each hour of the day and predict the leak node using the one corresponding to the time of measurements. The final output (the most likely leak node) is then calculated by majority vote:

$$y_t(\mathbf{x}) = \text{mode}(\widehat{y}_\tau(\mathbf{x})), \quad \tau = t-T,\ldots,t-1,t, \tag{6.25}$$

where $\text{mode}(\cdot)$ indicates the highest frequency class. In this way, the most predicted leak node is selected over the available time interval.

Figure 6.15 shows the result of an LDA leak localization test on the Hanoi network over a full day in 1-hour increments ($t = 1,2,\ldots,24$). The true leak node is number 8, and the output obtained by majority vote is number 9, although, at different times of the day the predicted node varies within the set $\{5,7,8,9,10\}$.



(a) Predicted node over time (b) Class frequencies

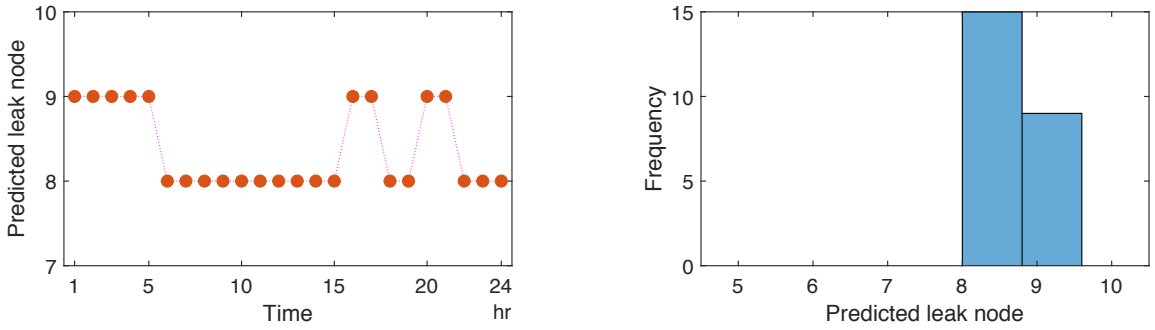
Fig. 6.15: Leak node prediction by majority vote over time

Another strategy to refine the output is, by considering the Bayesian inference described in Section 6.5, to take the posterior class probabilities at each sample time as prior probabilities for the next sample time. In this way, the prior and posterior probabilities are updated at each time step (with each new measurement) as

$$\widehat{P}_t(y_j \,|\, \mathbf{x}) = \frac{\widehat{P}_{t-1}(y_j \,|\, \mathbf{x})\, P(\mathbf{x} \,|\, y_j)}{P(\mathbf{x})}, \tag{6.26}$$

except at the initial time since there is still no posterior probability calculated. Therefore, the first prior probability is assigned as indicated in equations (6.14) and (6.15).

Figure 6.16 shows the results of the LDA leak localization test with the same test data as in Fig. 6.15 but by applying Bayesian inference over time. It can be noted that with this method the predicted node converges to the correct leak node and shows greater consistency (less variability) than the simple majority vote.



(a) Predicted node over time (b) Class frequencies

Fig. 6.16: Leak node prediction by Bayesian inference over time

It may be necessary to limit the probabilities to an interval $[0, 1-\varepsilon]$, where ε is a small number during the probability propagation over time. This prevents, when an estimated posterior probability is 1 at some sample time t_0, that the remaining probabilities are always 0 for subsequent time steps $t > t_0$.

6.10 Integrated leak localization method

The integrated procedure to locate leaks in water distribution networks, considering what has been described in previous sections, is shown in the following box. The workflow is the same for different classification methods and different feature sets.

Leak localization using classifiers

1. (**Data collection**) Using the network's hydraulic model, simulate different leakage scenarios, considering different leak flow rates and different leaky nodes. A nominal leak-free scenario should be simulated. EPANET software can be used in this step.
   From the simulation results, construct a pressure matrix, $\mathbf{P}$, containing the node pressures arranged by columns so that each row of this matrix represents a different leakage scenario. A matrix $\mathbf{P}^{(0)}$ (same size as $\mathbf{P}$) must also be constructed containing the leak-free node pressures. Additionally, a vector $\mathbf{y}$ must be constructed, which has a length equal to the number of rows in $\mathbf{P}$. This vector contains the class labels for the classifiers: the element y_i is an integer indicating the leak node for the leak scenario represented in the i-th row of $\mathbf{P}$.
2. (**Sensor placement**) From the pressure matrix $\mathbf{P}$, calculate an optimal sensor placement using the techniques proposed in Morales-González et al. (2021) and Santos-Ruiz et al. (2022). The number of sensors can be determined from a marginal analysis by progressively increasing the number of sensors until new sensors do not significantly improve leak localization performance. Some sensors less than the marginal number can be assigned depending on the available equipment.
3. (**Data partitioning**) In matrices $\mathbf{P}$ and $\mathbf{P}^{(0)}$, eliminate the columns that do not correspond to sensor nodes, keeping only the columns associated with the sensor placement calculated in the

previous step. Subsequently, separate the rows of $\mathbf{P}$ into two subsets: one for training and one for testing, creating the sub-matrices $\mathbf{P}_{\text{train}}$ and $\mathbf{P}_{\text{test}}$. The same division is performed on $\mathbf{P}^{(0)}$. Similarly, $\mathbf{y}$ is divided to obtain $\mathbf{y}_{\text{train}}$ and $\mathbf{y}_{\text{test}}$.

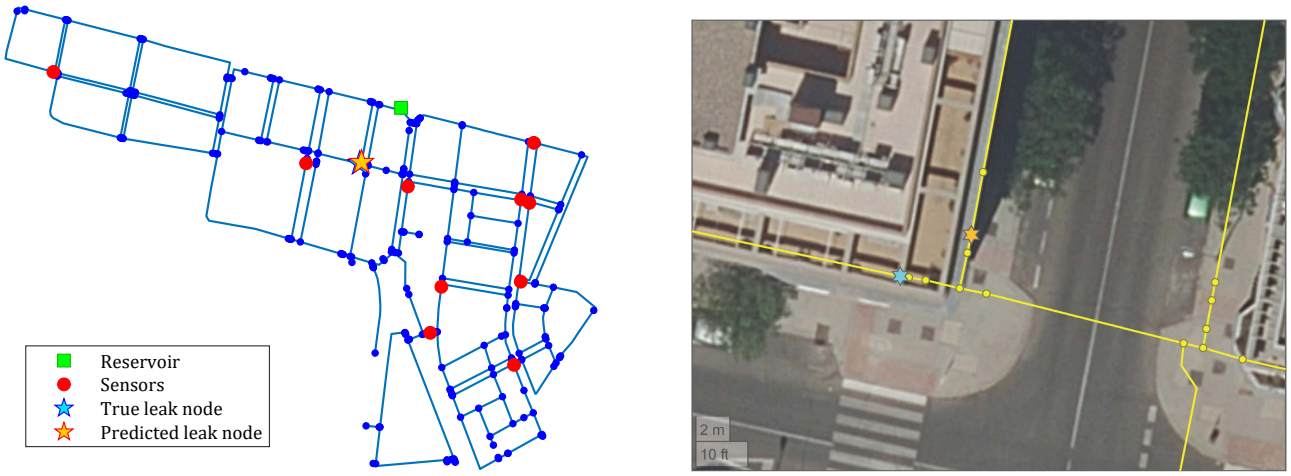
4. (**Feature computation**) Set the features for training $\mathbf{X}_{\text{train}} = \mathbf{P}_{\text{train}}$ or $\mathbf{X}_{\text{train}} = \mathbf{P}_{\text{train}} - \mathbf{P}^{(0)}_{\text{train}}$ or $\mathbf{X}_{\text{train}} = f\big(\mathbf{P}_{\text{train}} - \mathbf{P}^{(0)}_{\text{train}}\big)$, where $f(\cdot)$ is the cosine extractor function defined in (6.23). Similarly, set the features for testing: $\mathbf{X}_{\text{test}} = \mathbf{P}_{\text{test}}$ or $\mathbf{X}_{\text{test}} = \mathbf{P}_{\text{test}} - \mathbf{P}^{(0)}_{\text{test}}$ or $\mathbf{X}_{\text{test}} = f\big(\mathbf{P}_{\text{test}} - \mathbf{P}^{(0)}_{\text{test}}\big)$.
5. (**Classifier training**) Train the selected classifier using the training data labeled $\{\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}}\}$.
6. (**Classifier testing**) Test the classifier using the labeled test data $\{\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}}\}$. Store in a vector $\widehat{\mathbf{y}}$ the leak nodes predicted for $\mathbf{X}_{\text{test}}$. Then, evaluate the performance of the classifier from $\mathbf{y}_{\text{test}}$ and $\widehat{\mathbf{y}}$ by using the metrics described in Section 6.4.
   Steps 3 through 6 can be repeated in a cross-validation scheme. In that case, the K-fold loss is used as a performance measure.
7. (**On-line prediction**) Obtain new pressure measurements at the sensor nodes and compute the features in the same way as step 4. Use the classifier trained in step 5 to predict the leak's location. If multiple measurements are available and covering a time interval where the network's operating point changes considerably, use one of the proposals from Section 6.9 (majority vote or Bayesian inference) for the final estimate of the leak node.

The presented leak localization scheme was tested to evaluate the performance of different combinations of classifier and feature set. Table 6.4 summarizes the comparison of the performance of the different classifiers tested in the Hanoi network. In the tests, sensors were placed in three nodes (12, 21 and 27), and three different sets of features were considered: pressures, residuals and cosines. The results highlight the general improvement introduced by using cosine features instead of residuals. In addition, it is noted that k-NN with Euclidean distance and cosine features is equivalent to k-NN with cosine distance and residual features. It is also highlighted that QDA generally shows better performance than the other classifiers, except when cosine features are used, because in that case the performance of all the classifiers is quite close.

Table 6.4: Leak classification error for different classifiers on different feature sets. Error metric: 5-fold loss (cross-validation)

Classification Method	**Features**		
	Pressures	**Residuals**	**Cosines**
Naive Bayes	0.82968	0.81935	0.00645
Linear Discriminant Analysis	0.75161	0.76129	0.02000
Quadratic Discriminant Analysis	0.00645	0.00516	0.00839
Euclidean k-NN	0.48710	0.49548	0.00387
Cosine k-NN	0.67355	0.00387	0.00323
Decision Tree	0.70903	0.71161	0.00387

Using the work flow in the box above, leak localization tests were also performed in some sectors of the Madrid network (the DMA described in Section 6.3 and other medium-sized sectors) by using field measurements. An exhaustive analysis is not presented in these cases because the samples collected correspond only to a single leak point. Opening valves physically simulated the leaks. In each case, measurements were collected from 10 sensors placed in specific nodes determined by metaheuristic optimization, such as those described in Morales-González et al. (2021) and Santos-Ruiz et al. (2022). Nominal leak-free node pressures were obtained by simulation with an EPANET model tuned from measurements two days before the leak event. Figure 6.17 shows the result of the leak localization test in the DMA of the Madrid network. In this figure, the true leak node is represented by the cyan star (⋆) and the predicted leak node by the orange star (⋆). The topological distance between the true and predicted nodes (node number 272 and 138, respectively) is 5, corresponding to about 8 m of pipe length. The best leak node prediction (shown in Fig. 6.17) was obtained with k-NN using correlation distance. With k-NN using cosine distance and with other classifiers using cosine features, localization errors of about 100 m were obtained. This may be because the directionality associated with the directing cosines is sensitive both to noise in pressure measurements and to the accuracy of nominal pressure estimates obtained from the model. Therefore, poor leak-free pressure estimates can lead to fake directions associated with nodes distant from the leak node when the model is not well calibrated. In contrast, the correlation distance appears less sensitive to model calibration and measurement bias.



(a) Topological view (b) Close-up street view

Fig. 6.17: Leak localization in a DMA of the Madrid network using k-NN with correlation distance

In leak localization tests with synthetic data (simulations) in different sectors of the Madrid network, the same behavior was observed: the classifiers showed a considerably higher performance using cosine features than when they used raw residual features. In tests with physical measurements (not simulations), however, the difference was not always significant, and sometimes the k-NN with correlation distance produced better results.

6.11 Conclusions

This work has addressed a leak localization method using classifiers with transformed pressure residuals as features. The residuals' proposed transformation consists of a nonlinear mapping to extract the information about their directions, separating it from their magnitudes. The transformed features, called direction cosines, significantly improve leak localization accuracy compared with untransformed residuals, even under noisy pressure measurements. It should be noted that in predictions from real measurements in physical networks, a well-calibrated hydraulic model must be available to estimate the leak-free nominal pressures. The cosine distance and the directionality on which the cosine features are based strongly depend on the accuracy of those estimates. In contrast, the correlation distance seems less sensitive to model calibration and measurement bias, so using k-NN with correlation distance should be considered when the hydraulic model is not very accurate.

Two machine learning classification techniques were not explored in depth for leak localization and are not reported in this work: support vector machines (SVMs) and artificial neural networks (ANNs). SVMs are binary classifiers that cannot directly address the problem of classifying multiple leak classes, so their application involves using "one versus rest" or "one versus one" schemes, leading to a more complex leak localization in networks with many nodes. Furthermore, SVMs are linear classifiers, so they must be combined with kernel techniques to classify nonlinear decision boundaries. On the other hand, neural networks are a fairly broad field with an application that usually requires deep learning techniques not addressed in this work. In preliminary tests with SVMs and ANNs, no better performances were achieved than those reported with the other methods, but their exploration remains open for future work.

This page intentionally left blank

References

Akiba, T., Iwata, Y., and Yoshida, Y. (2013). Fast exact shortest-path distance queries on large networks by pruned landmark labeling. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 349–360.

Berglund, A., Areti, V. S., Brill, D., and Mahinthakumar, G. K. (2017). Successive Linear Approximation Methods for Leak Detection in Water Distribution Systems. *Journal of Water Resources Planning and Management*, 143(8):04017042.

Blocher, C., Pecci, F., and Stoianov, I. (2020). Localizing Leakage Hotspots in Water Distribution Networks via the Regularization of an Inverse Problem. *Journal of Hydraulic Engineering*, 146(4):04020025.

Box, G. (1988). Signal-to-Noise Ratios, Performance Criteria, and Transformations. *Technometrics*, 30(1):1–17.

Casillas, M. V., Garza-Castañón, L. E., and Puig, V. (2013). Extended-horizon analysis of pressure sensitivities for leak detection in water distribution networks: Application to the Barcelona network. In *2013 European Control Conference (ECC)*, pages 401–409.

Casillas, M. V., Garza-Castañón, L. E., and Puig, V. (2014). Model-based leak detection and location in water distribution networks considering an extended-horizon analysis of pressure sensitivities. *Journal of Hydroinformatics*, 16(3):649–670.

Casillas, M. V., Garza-Castañón, L. E., Puig, V., and Vargas-Martinez, A. (2015). Leak signature space: An original representation for robust leak location in water distribution networks. *Water*, 7(3):1129–1148.

De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D. (2000). The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1–18.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

Eliades, D. G., Kyriakou, M., Vrachimis, S., and Polycarpou, M. M. (2016). EPANET-MATLAB Toolkit: An Open-Source Software for Interfacing EPANET with MATLAB. In *Proc. 14th International Conference on Computing and Control for the Water Industry (CCWI)*, page 8, The Netherlands.

Ferrandez-Gamot, L., Busson, P., Blesa, J., Tornil-Sin, S., Puig, V., Duviella, E., and Soldevila, A. (2015). Leak localization in water distribution networks using pressure residuals and classifiers. *IFAC-PapersOnLine*, 48(21):220–225.

Fujiwara, O. and Khang, D. B. (1990). A two-phase decomposition method for optimal design of looped water distribution networks. *Water resources research*, 26(4):539–549.

Koutroumbas, K. and Theodoridis, S. (2008). *Pattern recognition*. Academic Press.

Martinez, W. L. and Martinez, A. R. (2015). *Computational Statistics Handbook with MATLAB*. Computer Science and Data Analysis Series. Chapman and Hall/CRC, 3 edition.

Morales-González, I., Santos-Ruiz, I., López-Estrada, F., and Puig, V. (2021). Pressure Sensor Placement for Leak Localization Using Simulated Annealing with Hyperparameter Optimization. In *2021 5th International Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 205–210. IEEE.

OECD (2016). Water Governance in Cities. *OECD Studies on Water*.

Pérez, R., Puig, V., Pascual, J., Peralta, A., Landeros, E., and Jordanas, L. (2009). Pressure sensor distribution for leak detection in Barcelona water distribution network. *Water Science and Technology: Water Supply*, 9(6):715–721.

Pérez, R., Puig, V., Pascual, J., Quevedo, J., Landeros, E., and Peralta, A. (2011). Methodology for leakage isolation using pressure sensitivity analysis in water distribution networks. *Control Engineering Practice*, 19(10):1157–1167.

Pérez, R., Quevedo, J., Puig, V., Nejjari, F., Cugueró, M. A., Sanz, G., and Mirats, J. M. (2011). Leakage isolation in water distribution networks: A comparative study of two methodologies on a real case study. In *2011 19th Mediterranean Conference on Control Automation (MED)*, pages 138–143.

Pérez, R., Sanz, G., Puig, V., Quevedo, J., Cugueró, M. À., Nejjari, F., Meseguer, J., Cembrano, G., Mirats, J. M., and Sarrate, R. (2014). Leak localization in water networks: A model-based methodology using pressure sensors applied to a real network in Barcelona [applications of control]. *IEEE Control Systems Magazine*, 34(4):24–36.

Pilcher, R., Hamilton, S., Chapman, H., Ristovski, B., and Strapely, S. (2007). Leak Location and Repair Guidance Notes. *International Water Association—Water Loss Task Forces: Specialist Group Efficient Operation and Management, Bucharest, Romania*.

Pudar, R. S. and Liggett, J. A. (1992). Leaks in Pipe Networks. *Journal of Hydraulic Engineering*, 118(7):1031–1046.

Puig, V., Ocampo-Martínez, C., Pérez, R., Cembrano, G., Quevedo, J., and Escobet, T., editors (2017). *Real-time Monitoring and Operational Control of Drinking-Water Systems*. Springer International Publishing.

Rossman, L. A., Woo, H., Tryby, M., Shang, F., Janke, R., and Haxton, T. (2020). EPANET 2.2 User Manual. Technical Report EPA/600/R-20/133, U.S. Environmental Protection Agency, Washington, DC.

Santos-Ruiz, I., López-Estrada, F.-R., Puig, V., Valencia-Palomo, G., and Hernández, H.-R. (2022). Pressure Sensor Placement for Leak Localization in Water Distribution Networks Using Information Theory. *Sensors*, 22(2).

Sanz, G. and Pérez, R. (2015). Sensitivity Analysis for Sampling Design and Demand Calibration in Water Distribution Networks Using the Singular Value Decomposition. *Journal of Water Resources Planning and Management*, 141(10):04015020.

Sanz, G., Pérez, R., Kapelan, Z., and Savic, D. (2016). Leak Detection and Localization through Demand Components Calibration. *Journal of Water Resources Planning and Management*, 142(2):04015057.

Young, E. C. (2017). *Vector and Tensor Analysis*. Pure and Applied Mathematics. CRC Press, 2 edition.