Universitat Politècnica de Catalunya

Programa de Doctorat:

Automàtica, Robòtica i Visió

Tesi Doctoral:

Robust Leak Localization in Water Distribution Networks Using Machine Learning Techniques

Adrià Soldevila Coma

Directors de la tesi:
Vicenç Puig Cayuela i Sebastián Tornil Sin

Febrer del 2018

Acknowledgements

I would like to thank all the members of the researcher Center for Supervision, Safety and Automatic Control (CS^2AC) group, the Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAII) and Universitat Politècnica de Catalunya (UPC) for their support and friendship these three years, since it has been a great place to work.

During these three years I had the opportunity to do an internship in the POLitecnico de MIlano (POLIMI) in which I would like to thank professors Giacomo Boracchi and Manuel Roveri and all the members from the Dipartimento in Elettronica, Informazione e Bioengegneria.

Also, I want to thank my family and my friends for all of their support during this time which it has been very enthusiastic.

Finally, I would like to thank the team with I develop this PhD thesis, Joaquim Blesa and Rosa Maria Fernández, and specially to my directors Vicenç Puig and Sebastián Tornil for their trust in me.

This PhD thesis has been founded by the Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR), the European Social Fund (ESF) and the Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement de la Generalitat de Catalunya through the grant FI-DGR 2015 (ref. 2015 FI_B 00591).

Abstract

This PhD thesis presents a methodology to detect, estimate and locate water leaks (with the main focus in the localization problem) in water distribution networks using hydraulic models and machine learning techniques. The actual state of the art is introduced, the theoretical basis of the machine learning techniques applied are explained and the hydraulic model is also detailed. The whole methodology is presented and tested into different water distribution networks and district metered areas based on simulated and real case studies and compared with published methods.

The focus of the contributions is to bring more robust methods against the uncertainties that affects the problem of leak detection, by dealing with them using the self-similarity to create features monitored by the change detection technique intersection-of-confidence-interval, and the leak localization where the problem is tackled using machine learning techniques. By using those techniques, it is expected to learn the leak behavior considering their uncertainty to be used in the diagnosis stage after the training phase.

One method for the leak detection problem is presented that is able to estimate the leak size and the time instant when the leak has been produced. This method captures the normal, leak-free, behavior and contrast it with the new measurements in order to evaluate the state of the network. If the behavior is not normal, it is checked if it is due to a leak. To have a more robust leak detection method, a specific validation is designed to operate specifically with leaks and in the temporal region where the leak is most apparent.

The proposed technique is compared with other published methods providing a more reliable detection, specially with small leaks, as long as more information can be used later in the leak localization stage to improve it but at the cost of slower detection time than the other methods.

A methodology to extend the current model-based approach to localize water leaks by means of classifiers is proposed where the non-parametric k-nearest neighbors classifier and the parametric multi-class Bayesian classifier are proposed.

The proposed model-based leak localization using classifiers allows to better handle the uncertainty surrounding the data used for the diagnosis which derives in an improved precision of the localization result. The main drawback relies on the computational cost, in an off-line stage, of the data required by the classifier to learn the dispersion of the data. Also, the method is highly dependent of the hydraulic model of the network.

A new data-driven approach to localize leaks using a multivariate regression technique without the use of hydraulic models is also introduced. This method presents a clear benefit over the model-based techniques by removing the need of the hydraulic model despite of topological information is still required. Also, the information of the expected leaks is not required since information of the expected hydraulic behavior with leak is exploited to find the place where the leak is more probable. This method has a good performance in practice, but it is very sensitive to the number of sensors in the network and their location.

The performance of leak localization methods is highly sensitive to the sensor placement. Additionally, it must be noticed that the optimal for different leak localization methods can be different. With the aim of maximizing the performance of the proposed leak localization methods, several sensor placement approaches are presented and evaluated since the combinatorial problem can not be handled by trying each possible sensor configuration except for the smallest networks with only few sensors to install. The proposed approaches exploit the potential of feature selection techniques to perform the desired sensor placement task.

The proposed sensor placement techniques reduce the computational load required to

take into account the amount of data needed to model the uncertainty compared with other optimization approaches while are designed to work with the leak localization problem. More precisely, the proposed hybrid feature selection technique for sensor placement is able to work with any leak localization method that can be evaluated using a confusion matrix and still being optimum for that method. This last method is good for a few sensors, but lacks of precision when the number of sensors to place is large. To overcome this problem, an incremental sensor placement is proposed which is better for a larger number of sensors to place but worse when the number is small.

Resum

Aquesta tesi presenta una nova metodologia per a localització de fuites en xarxes de distribució d'aigua potable. Primer s'ha revisat l'estat del art actual i les bases teòriques tant de les tècniques de machine learning utilitzades al llarg de la tesi com els mètodes existents per a la localització de fuites. La metodologia presentada s'ha provat en diferents xarxes d'aigua simulades i reals, comparant el resultats amb altres mètodes publicats.

L'objectiu principal de la contribució aportada és el de desenvolupar mètodes més robustos enfront les incerteses que afecten a la localització de fuites. En el cas de la detecció i estimació de la magnitud de la fuita, s'utilitza la tècnica self-similarity per crear els indicadors que es monitoritzen amb la tècnica de detecció de canvis ("intersection-of-confidence-intervals"). En el cas de la localització de les fuites, s'han fet servir les tècniques de classificadors i interpoladors provinents del machine learning. A l'utilitzar aquestes tècniques s'espera captar el comportament de la fuita i de la incertesa per aprendre i tenir-ho en compte en la fase de la localització de la fuita.

El mètode de la detecció de fallades proposat és capaç d'estimar la magnitud de la fuita i l'instant en que s'ha produït. Aquest mètode captura el comportament normal, sense fuita, i el contrasta amb les noves mesures per avaluar l'estat de la xarxa. En el cas que el comportament no sigui el normal, es procedeix a comprovar si això és degut a una fuita. Per tenir una mètode de detecció més robust, es fa servir una capa de validació especialment dissenyada per treballar específicament amb fuites i en la regió temporal en que la fuita és més evident.

La tècnica proposada s'ha comparat amb altres mètodes ja publicats donant com a resultat una detecció més fiable, especialment en el cas de les fuites més petites. Al mateix temps dona més informació que pot ser utilitzada després en la fase de localització de la fuita per tal de millorar-la. L'únic inconvenient d'aquest mètode, és que és més lent en la detecció que els altres mètodes analitzats.

Per tal de millorar l'actual metodologia de localització de fuites mitjançant models hidràulics s'ha proposat l'ús de classificadors. Per una banda es proposa el classificador no paramètric k-nearest neighbors i per l'altre banda el classificador Bayesià paramètric per múltiples classes.

Les tècniques de localització de fuites basades en models i classificadors permeten controlar millor la incertesa associada a les dades utilitzades en la diagnosi que resulta en una millora de la precisió en la localització de la fuita. El principal problema recau en el cost computacional, tot i que no cal que és realitzi en temps real, de les dades necessàries per entrenar el classificador de forma que aprengui bé la dispersió de les dades. També, destacar que els mètodes són molt dependents del model hidràulic que descriu la xarxa.

En el camp de la localització de fuites, també s'ha desenvolupat un nou mètode de localització de fuites basat en models de dades mitjançant la regressió de múltiples paràmetres sense l'ús del model hidràulic de la xarxa. Aquest mètode presenta uns beneficis clars respecte a les tècniques basades en models hidràulics a l'eliminar el model hidràulic tot i que la informació topològica encara és necessària. També representa un avantatge el fet que la informació de cada fallada contemplada no és necessària, ja que s'utilitza el coneixement hidràulic que s'espera quan hi ha una fuita en un determinat lloc de la xarxa. Aquest mètode ha donat bons resultats a la pràctica, però és especialment sensible al nombre de sensors utilitzats i a la seva disposició en la xarxa.

Finalment, s'ha tractat el problema de la col·locació de sensors. El rendiment de la localització de fuites està relacionada amb la col·locació de sensors i és particular per a cada mètode de localització. Amb l'objectiu de maximitzar el rendiment dels mètodes de localització de fuites presentats anteriorment, es presenten i aval-

uen tècniques de col·locació de sensors específicament dissenyats ja que el problema combinatori no es pot tractar avaluant cada possible combinació de sensors excepte en les xarxes més petites amb pocs sensors per instal·lar. Aquestes tècniques de col·locació de sensors exploten el potencial de les tècniques de selecció de variables per tal de realitzar la tasca desitjada.

Les tècniques proposades de col·locació de sensors redueixen el cost computacional necessari, al tenir en compte les dades necessàries per tal de modelar la incertesa, comparat amb altres tècniques d'optimització al mateix temps que està dissenyat per treballar específicament per la tasca de localització de fuites. Més concretament, la proposta de la tècnica híbrida de selecció de variables és capaç de treballar amb qualsevol mètode de localització de fuites que és pugui avaluar amb la matriu de confusió de tal forma que continuarà sent òptim per a aquesta tècnica. Aquest mètode té un bon comportament quan s'han de col·locar pocs sensor, però li falta precisió quan el nombre de sensors a col·locar augmenta. Per resoldre aquest problema, s'ha proposat mètode incremental de col·locació de sensors que té millors resultats en cas d'un gran nombre de sensors a col·locar però pitjor quan el nombre de sensors és petit.

Resumen

Esta tesis doctoral presenta una nueva metodología para detectar, estimar el tamaño y localizar fugas de agua (donde el foco principal está puesto en el problema de la localización de fugas) en redes de distribución de agua potable. La tesis presenta una revisión de el estado actual y las bases de las técnicas de machine learning que se aplican así como una explicación del modelo hidráulico de las redes de agua. El conjunto de la metodología se presenta y prueba en diferentes redes de distribución de agua y sectores de consumo con casos de estudio simulados y reales, y se compara con otros métodos ya publicados.

La contribución principal es la de desarrollar métodos más robustos frente a la incertidumbre de los datos. En la detección de fugas, la incertidumbre se trata con la técnica del self-similarity para la generación de indicadores que luego son monitoreados per la técnica de detección de cambios conocida como intersection-of-confidece-interval. En la localización de fugas el problema de la incertidumbre se trata con técnicas de machine learning. Al utilizar estas técnicas se espera aprender el comportamiento de la fuga y su incertidumbre asociada para tenerlo en cuenta en la fase de diagnóstico.

El método presentado para la detección de fugas tiene la habilidad de estimar la magnitud y el instante en que la fuga se ha producido. Este método captura el comportamiento normal, sin fugas, del sistema y lo contrasta con las nuevas medidas para evaluar el estado actual de la red. En el caso de que el comportamiento no sea el normal, se comprueba si es debido a la presencia de una fuga en él sistema. Para obtener un método de detección más robusto, se considera una capa de validación

especialmente diseñada para trabajar específicamente con fugas y durante el periodo temporal donde la fuga es más evidente.

Está técnica se compara con otras ya publicadas proporcionando una detección más fiable, especialmente en el caso de fugas pequeñas, al mismo tiempo que proporciona más información que puede ser usada en la fase de la localización de la fuga permitiendo mejorarla. El principal problema es que el método es más lento que los otros métodos analizados.

Con el fin de mejorar la actual metodología de localización de fugas mediante modelos hidráulicos, se propone la utilización de clasificadores. Concretamente, se propone el clasificador no paramétrico k-nearest neighbors y el clasificador Bayesiano paramétrico para múltiples clases.

La propuesta de localización de fugas mediante modelos hidráulicos y clasificadores permite gestionar la incertidumbre de los datos mejor para obtener un diagnóstico de la localización de la fuga más preciso. El principal inconveniente recae en el coste computacional, aunque no se realiza en tiempo real, de los datos necesarios por el clasificador para aprender correctamente la dispersión de los datos. Además, el método es muy dependiente de la calidad del modelo hidráulico de la red.

En el campo de la localización de fugas, se a propuesto un nuevo método de localización de fugas basado en modelos de datos mediante la regresión de múltiples parámetros sin el uso de modelo hidráulico. Este método presenta un claro beneficio respecto a las técnicas basadas en modelos hidráulicos ya que prescinde de su uso, aunque la información topológica de la red es aún necesaria. Además, la información del comportamiento de la red para cada fuga no es necesario, ya que el conocimiento del efecto hidráulico de una fuga en un determinado punto de la red es utilizado para la localización. Este método ha dado muy buenos resultados en la práctica, aunque es muy sensible al número de sensores y a su colocación en la red.

Finalmente, se trata el problema de la colocación de sensores. El desempeño de la localización de fugas está ligado a la colocación de los sensores y es particular para cada método. Con el objetivo de maximizar el desempeño de los métodos de localización de fugas presentados, técnicas de colocación de sensores específicamente diseñados

para ellos se han presentado y evaluado. Dado que el problema de combinatoria que presenta no puede ser tratado analizando todas las posibles combinaciones de sensores excepto en las redes más pequeñas con unos pocos sensores para instalar. Estas técnicas de colocación de sensores explotan el potencial de las técnicas de selección de variables para realizar la tarea deseada.

Las técnicas de colocación de sensores propuestas reducen la carga computacional, requerida para tener en cuenta todos los datos necesarios para modelar bien la incertidumbre, comparado con otras propuestas de optimización al mismo tiempo que están diseñadas para trabajar en la tarea de la localización de fugas. Más concretamente, la propuesta basada en la técnica híbrida de selección de variables para la colocación de sensores es capaz de trabajar con cualquier técnica de localización de fugas que se pueda evaluar con la matriz de confusión y ser a la vez óptimo. Este método es muy bueno para la colocación de sensores pero el rendimiento disminuye a medida que el número de sensores a colocar crece. Para evitar este problema, se propone método de colocación de sensores de forma incremental que presenta un mejor rendimiento para un número alto de sensores a colocar, aunque no es tan eficaz con pocos sensores a colocar.

Notation

The following notation is used along the PhD thesis.

Variables, Vectors and Matrices

a: exponential flow coefficient

$\mathbf{c}$, $\tilde{\mathbf{c}}$ and $\bar{\mathbf{c}}$: actual, measured and generated boundary conditions

$\mathbf{c}_d$: operational conditions

cl: class

d_i, $\hat{d}_i$ and $\bar{d}_i$: actual, estimated and generated demand at node i

$d_i^{(u)}$: amplitude of nodal demand uncertainty at node i

$\tilde{d}_{WDN}$, $\hat{d}_{WDN}$ and $\bar{d}_{WDN}$: measured, estimated and generated global consumption

$\tilde{d}_{WDN}^{(T)}$: reference global consumption

$d_{WDN}^{(u)}$: amplitude of global demand consumption uncertainty

e: squared relative error

$\mathbf{e}^{(H)}$: vector with historical squared relative errors

e_c: elite count value

f_i: sum of flow that pass through node i

$fl_{i,j}$: flow between nodes i and j

$fun(\cdot)$: function

$fun_p(\cdot)$: polynomial function

$fun_c(\cdot)$: correlation function

h_i: head (pressure taking into account the geodesic level) at node i

k: number of nearest neighbors used in the k-Nearest Neighbors algorithm

l_i, $\hat{l}_i$ and $\bar{l}_i$: actual, estimated and generated leak at node i

$l_i^{(u)}$: amplitude of leak size uncertainty at node i

$l^{(0)}$: leak without uncertainty

l_{min}: minimum detectable leak

Δl: leak size estimation error

m: dimension of the variagrom in the Kriging interpolation

m_T: samples in the training data set for each class

m_V: samples in the validation data set for each class

max_g: maximum number of generations

n_b: number of boundary conditions

n_c: number of classes

n_f: number of features

$n_f^{(R)}$: reduced number of features

n_n: number of nodes

n_s: number of measurements

n_{sp}: number of sensor already placed

p_i, $\hat{p}_i$ and $\bar{p}_i$: actual, estimated and generated pressure at node i

$\hat{p}_i^{(0)}$ and $\hat{p}_i^{(l_j,0)}$: estimated pressure without leak and with leak and both without uncertainties

p_s: population size

$\mathbf{q}$: vector with the sensor configuration

$\mathbf{q}^{(H)}$: vector of historical sensor configurations for each number of sensors

$\mathbf{r}$: vector of residuals

$\mathbf{r}^{(+)}$: vector of processed (to make them all positive) residuals

$\mathbf{s}$: current self-similarity patch

t: continuous time

tol: stopping tolerance criteria

u: order of the polynomial function in the Kriging interpolation

$\mathbf{w}$: vector of values used in the Wilcoxon's test

$\mathbf{x}$: vector of inputs (to a machine learning technique)

$\mathbf{x}_T$: vector of training inputs (to a machine learning technique)

$\mathbf{y}$: vector of outputs (from a machine learning technique)

$\mathbf{y}_T$: vector of training outputs (from a machine learning technique)

C: roughness of the pipe

D: diameter of the pipe

$\mathbf{D}$: minimum topological distance matrix

$\mathbf{F}$: feature space

$\mathbf{F}^{(R)}$: reduced feature space

$\mathbf{I}$: matrix of population members in the genetic algorithm

L: length of the pipe

N: time horizon

Nr: number of valid samples in the Wilcoxon's test

P: probability

$\mathbf{Q}$: matrix with different sensor configurations in the genetic algorithm

$\mathbf{Q}^{(H)}$: matrix with historic sensor configurations for each number of sensors in the genetic algorithm

$\mathbf{R}$: vector of relevance values of each feature

$\mathbf{R}_R$: vector of relevance values ranked starting with the most relevant

$\mathbf{T}$: training data set

$\mathbf{T}^{(R)}$: reduced training data set

T^*: leak starting time

$\widehat{T}^*$: estimated leak starting time

$\widehat{T}$: detection time

$\mathbf{V}$: validation data set

$\mathbf{V}^{(R)}$: reduced validation data set

W: Wilcoxon's test result before the application of the reference tables to obtain the test statistic

α: demand pattern coefficient

β: size of the repetitive pattern

γ: user defined threshold

δ: number of samples used to validate a detected change

$\varepsilon(\cdot)$: function part of the Kriging interpolation

$2\zeta + 1$: self-similarity patch size

$\boldsymbol{\theta}$: vector of correlation parameters for the Kriging interpolation

ϑ: user defined threshold

ι: number of samples for the creation of a confidence interval

$\boldsymbol{\lambda}_i$: i^{th} row of the average training matrix

μ: mean of the current self-similarity features

μ_T: mean of the self-similarity features training data set

ν and $\bar{\nu}$: actual and generated noise

$\nu^{(u)}$: amplitude noise uncertainty

ξ: indicator of the position of patch centers inside the self-similarity training data set

$\boldsymbol{\pi}$: most similar patch to the current one inside the self-similarity training data set

ϖ: estimated parameter in the Kriging interpolation

ρ: Pearson's correlation coefficient

ϱ: self-similarity feature

σ: standard deviation of the current self-similarity features

σ_T: standard deviation of the self-similarity features training data set

ς: estimated parameter in the variable part in the Kriging interpolation

τ: estimated parameter for the constant part in the Kriging interpolation

φ: user defined threshold based on $\mathfrak{y}$ value

$\boldsymbol{\chi}$: vector of polynomial coefficients for the Kriging interpolation

Γ: confusion matrix

$\boldsymbol{\Lambda}$: average training matrix

Υ: confidence parameter for the creation of intervals-of-confidence

$\boldsymbol{\Phi}$: matrix with the correlation distance between each pair of features

$\boldsymbol{\Phi}^{(B)}$: matrix with the allowed pairs of combinations

Ψ: data distribution

$\boldsymbol{\Omega}$: sensitivity matrix

$\mathcal{I}$: interval of confidence

$\mathcal{I}^{(+)}$: upper bound of a confidence interval

$\mathcal{I}^{(-)}$: lower bound of a confidence interval

$\mathcal{R}$: vector with the weight of the samples used in the Wilcoxon's test

$\mathfrak{a}_i$: discrepancy between the simulated and real measurement i

$\mathfrak{a}_i^{(c)}$: centered to zero discrepancy between the simulated and real measurement i

$\mathfrak{d}_i$: i^{th} row of the matrix $\mathfrak{D}$

$\mathfrak{f}$: position of the actual measurement inside the repetitive pattern considered

$\mathfrak{r}$: coefficient of the pipe resistance

$\mathfrak{s}$: i^{th} row of the sensitivity matrix

$\mathfrak{y}$: average value of $\mathbf{\Phi}$ matrix without the diagonal

$\mathfrak{A}_i$: vector of measurements of attribute i for a time period

$\mathfrak{D}$: matrix with the minimum pipe distance between nodes

$\mathfrak{D}_s$: submatrix of $\mathfrak{D}$ with the minimum pipe distance between nodes with sensors installed

$\mathfrak{H}$: constant part of the Kriging interpolation

$\mathfrak{S}$: normalized $\mathbf{r}^{(+)}$

$\mathfrak{S}_{max}$: maximum possible value of $\mathfrak{S}$

$\mathfrak{S}_{min}$: minimum possible value of $\mathfrak{S}$

$\mathfrak{Z}$: user defined parameter in the residual normalization process

Acronyms

Ac: Accuracy

AIS: Artificial Immune System

AL: Acoustic Logging

AMR: Automatic Meter Reader

ANN: Artificial Neural Network

ApEn: Approximate Entropy

ARX: AutoRegressive with eXogenous terms

BFS: Backward Feature Selection

BN: Bayesian Network

BR: Bayesian Reasoning

CCM: Consumers Contact Model

CDT: Change Detection Test

CFPD: Comparison of Flow Pattern Distributions

CPS: Cyber Physical System

CUSUM: CUmulative SUMmation

DD: Detection Delay

DMA: District Metered Area

DMOD: Distance to MODel

DPCA: Distributed Principal Component Analysis

DTD: Difference Time Detection

EA: Evolutionary Algorithm

EM: Expectation Maximization

EN: Elman Network

EPR: Evolutionary Polynomial Regression

ER: Evidential Reasoning

ES: Exhaustive Search

FCBF: Fast Correlation-Based Filter

FDI: Fault Detection and Isolation

FFT: Fast Fourier Transform

FNR: False Negative Rate

FPR: False Positive Rate

FRM: Frequency Response Analysis

FS: Feature Selection

FSM: Fault Signature Matrix

GA: Genetic Algorithm

GMM: Gaussian Mixture Model

GPR: Ground Penetrating Radar

HLR: Head Loss Ratio

HMM: Hidden Markov Model

IA: Inverse Analysis

ICI: Intersection of Confidence Interval

IDW: Inverse Distance Weighted interpolation

i.i.d.: independent and identically distributed

IRA: Inverse Response Analysis

ITA: Inverse Transient Analysis

JTFA: Joint Time frequency Analysis

KF: Kalman Filter

k-NN: k-Nearest Neighbors

KPCA: Kernel Principal Component Analysis

LNC: Leak Noise Correlator

LP: Local Polynomial interpolation

LPV: Linear Parameter Varying

LRM: Leak Reflection Method

MDN: Mixture Density Model

MLP: Multi-Layer Perception

MNF: Minimum Night Flow

NKF: Non-linear Kalman Filter

NOP: Normal Operating Patterns

NN: Nearest Neighbor

NPW: Negative Pressure Wave

NRW: Non-Revenue Water

OC: Ordinary Cokriging

OK: Ordinary Kriging

PBMP: Pipe Burst Model Prediction

PCA: Principal Component Analysis

PDF: Probability Density Function

PF: Particle Filter

PGM: PiG-Mounted acoustic

PRV: Pressure Reducing Valve

PSO: Particle Swarm Optimization

RBF: Radial Basis Function

RF: Random Forest

SCEM-UA: Shuffled Complex Evolution Metropolis

SFFS: Sequential Forward Floating Selection

SOM: Self Organized Map

SPC: Statistical Process Control

SPE: Squared Prediction Error

SS: Self-Similarity

STFT: Short-Time Fourier Transform

SVD: Single Value Decomposition

SVM: Support Vector Machines

SVR: Support Vector Regression

SWDM: Standing Wave Difference Method

TGT: Tracer Gas Technique

WDN: Water Distribution Network

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Objectives 2

1.3 Outline of the PhD Thesis 3

1.3.1 Chapter 2: Background 3

1.3.2 Chapter 3: Case Studies 4

1.3.3 Chapter 4: Leak Detection 4

1.3.4 Chapter 5: Model-Based Leak Localization 4

1.3.5 Chapter 6: Data-Driven Leak Localization 5

1.3.6 Chapter 7: Sensor Placement 6

1.3.7 Chapter 8: Conclusions 6

2 Background **7**

2.1 Water Distribution Networks 7

2.1.1 Modeling the Leak as an Extra Demand 9

2.1.2 Uncertainty Modeling Using Artificial Data 10

2.1.3 Uncertainty Modeling with Real Data 12

2.2 Leak Detection and Localization 13

2.2.1 Leak Assessment 14

2.2.2 Step Testing 14

2.2.3 Acoustic and Vibration Techniques 15

2.2.4 Surface Analyzer Methods 16

2.2.5 Hydraulic Analysis . . . 18
2.2.6 Analysis Using Statistical Methods . . . 26
2.2.7 Analysis Using Machine Learning . . . 29
2.2.8 Summary of Leak Detection and Localization Techniques . . . 36
2.2.9 Sensor Placement Techniques . . . 36
2.3 Validation and Evaluation Indicators . . . 39
2.3.1 Leak Detection Indicators . . . 39
2.3.2 Leak Localization Indicators . . . 40
2.4 Machine Learning . . . 41
2.4.1 Basic Terminology . . . 42
2.4.2 Kinds of Techniques and Learning Methods . . . 42
2.4.3 Semi-Supervised Learning . . . 43
2.4.4 Supervised Learning . . . 45
2.4.5 Feature Selection . . . 50
2.4.6 Optimization . . . 54

3 Case Studies **56**
3.1 Barcelona DMAs . . . 56
3.1.1 Bellamar DMA . . . 57
3.1.2 Canyars DMA . . . 58
3.1.3 Parc de la Muntanyeta DMA . . . 58
3.1.4 Gavà Centre DMA . . . 59
3.1.5 Can Roca DMA . . . 60
3.2 Hanoi WDN . . . 61
3.3 Limassol DMA . . . 62
3.4 Nova Icària DMA . . . 62
3.4.1 Real Case . . . 63
3.5 Pavones DMA . . . 65
3.5.1 Real Case . . . 65

4 Leak Detection **67**

4.1 Leak Detection Scheme 69

4.2 First Layer: Detection 69

4.2.1 Leak Starting Time Estimation 72

4.3 Second Layer: Validation 73

4.3.1 Wilcoxon's Test 74

4.3.2 Leak Size Estimation 74

4.4 Case Study 74

5 Model-Based Leak Localization **80**

5.1 Principle of Model-Based Leak Localization Approaches 81

5.2 Limitations of Sensitivity Analysis Approaches 82

5.3 Basic Architecture and Operation 84

5.4 Methodology Overview 85

5.4.1 Off-line Stage 85

5.4.2 On-line Stage 88

5.5 k-NN Classifier Implementation 88

5.5.1 The k-NN Classifier 88

5.5.2 Time Reasoning 89

5.6 Bayesian Classifier Implementation 89

5.6.1 Bayesian Classification 89

5.6.2 Recursivity 90

5.6.3 Bayesian Time Reasoning 91

5.6.4 Calibration of the Probability Density Functions 91

5.7 Case Studies 92

5.7.1 Hanoi WDN Case Study 93

5.7.2 Nova Icària DMA Case Study 101

6 Data-Driven Leak Localization **104**

6.1 Assumptions and Basic Operation 105

6.1.1 Pressure Estimation by Kriging Interpolation 107

6.1.2 Bayesian Time Reasoning . . . 108
6.1.3 Summary . . . 110
6.2 Case Studies . . . 111
6.2.1 Hanoi WDN Case Study . . . 111
6.2.2 Nova Icària DMA Case Study . . . 114
6.2.3 Pavones DMA Case Study . . . 115

7 Sensor Placement **117**
7.1 Optimal Sensor Placement for Classifiers . . . 119
7.1.1 Sensor Placement Using Genetic Algorithms . . . 119
7.1.2 Sensor Distance Matrix . . . 120
7.1.3 Data Format . . . 121
7.1.4 Hybrid Feature Selection Approach . . . 125
7.2 Incremental Feature Selection Approach . . . 133
7.3 Case Studies . . . 136
7.3.1 Hanoi WDN Case Study . . . 138
7.3.2 Limassol DMA Case Study . . . 150

8 Conclusions and Future Work **158**
8.1 Conclusions . . . 158
8.1.1 Leak Detection . . . 158
8.1.2 Model-Based Leak Localization . . . 158
8.1.3 Data-Driven Leak Localization . . . 159
8.1.4 Sensor Placement . . . 159
8.2 Future Work . . . 160
8.2.1 Leak Detection . . . 160
8.2.2 Model-Based Leak Localization . . . 161
8.2.3 Data-Driven Leak Localization . . . 161
8.2.4 Sensor Placement . . . 161

List of Figures

2.1 Limassol topological DMA network. 8
2.2 Example of the procedures of change detection (black intervals) and estimated change starting time (blue intervals). 46
2.3 The k-NN algorithm. 47
2.4 Example of the fast correlation-based filter. 52
2.5 Sequential Forward Floating Selection. 53
2.6 Genetic algorithm. 55

3.1 Bellamar DMA network and flow measurements. 57
3.2 Canyars DMA network and flow measurements. 58
3.3 Parc de la Muntanyeta DMA network and flow measurements. . . . 59
3.4 Gavà Centre DMA network and flow measurements. 60
3.5 Can Roca DMA network and flow measurements. 61
3.6 Hanoi WDN. 61
3.7 Limassol DMA network. 62
3.8 Nova Icària DMA network. 63
3.9 Nova Icària real case measurements. 64
3.10 Pavones DMA network and their sensor placement. 65
3.11 Pavones DMA real case measurements. 66

4.1 Leak detection and starting time estimation scheme. 68
4.2 Proposed methodology. 69

4.3 Self-similarity ϱ feature in Ballamar DMA for the three leak cases, the vertical black line marks the leak starting time. 79

5.1 Leak localization scheme. 85
5.2 Residual data generation scheme. 87
5.3 PDF calibration for leaks 1 and 2. 92
5.4 Hanoi topological WDN and their sensor placement. 93
5.5 Residual space without any uncertainties in Hanoi WDN (each color represents a leak in a different location). 94
5.6 Residual space with uncertainties and noise in Hanoi WDN (each color represents a leak in a different location). 95
5.7 Original and artificially generated daily consumption. 96
5.8 Residual space with uncertainties in Hanoi WDN (each color represents a leak in a different location). 97
5.9 Accuracy results over a time horizon for the k-NN classifier in Hanoi WDN. 99
5.10 Accuracy results over a time horizon for the Bayesian classifier in Hanoi WDN. 100
5.11 Accuracy results over a time horizon for the Angle method in Hanoi WDN. 100
5.12 Average topological distance results in a time horizon in Hanoi WDN. 101
5.13 Comparison of different leak localization methods in Nova Icària DMA. 103

6.1 Normalized sensitivity matrix of the Hanoi WDN for a leak of 100 [l/s] with all the sensors. 112
6.2 Kriging interpolation for the case of a leak of 100 [l/s] at node 16. . 112
6.3 ATD results of the proposed leak localization approach in the Hanoi WDN. 114
6.4 Nova Icària DMA leak localization results. 115
6.5 Pavones DMA leak localization results. 116

7.1 Scheme of optimization process. 120

7.2 Data format. 122
7.3 Hybrid feature selection scheme. 126
7.4 Pressure data generation scheme. 134
7.5 Accuracy curves for both classifiers using the sensor placements obtained using GA with the objective to maximize the Ac in Hanoi WDN. 142
7.6 Sensor placement for the k-NN and Bayesian classifiers (same sensors) in Hanoi WDN. 143
7.7 Sensor placement and the ATD curves using the hybrid feature selection for both classifiers in Hanoi WDN. 147
7.8 ATD performance for the incremental feature selection sensor placement using different number of sensors in the Hanoi WDN. 148
7.9 Normalized sensitivity matrix of the Hanoi WDN for a leak of 100 [l/s] for some sensor placements by the incremental feature selection method. 149
7.10 Sensor placement using incremental feature selection for six sensors in the simplified Hanoi topological WDN. 149
7.11 Example of a daily flow consumption in the Limassol DMA network. 150
7.12 Accuracy curves for both classifiers using the sensor placements obtained using GA with the objective to maximize the Ac in Limassol DMA network. 153
7.13 Sensor placement for the k-NN and Bayesian classifiers in the Limassol DMA network. 154
7.14 Sensor placement and the ATD curves using the hybrid feature selection for both classifiers in Limassol DMA network. 157

List of Tables

2.1 Confusion matrix Γ. 40

4.1 Different artificial leaks injected into the Barcelona DMAs in [l/s]. . 75
4.2 Comparison of leak detection performance with small leaks. 78
4.3 Comparison of leak detection performance with large leaks. 78
4.4 Comparison of leak detection performance with bursts. 78

5.1 Accuracy results in Hanoi WDN. 98

6.1 Leak localization results in the simplified Hanoi WDN. 113

7.1 Sensor placement results using ES and GA in the Hanoi WDN for the k-NN classifier. 140
7.2 Sensor placement results using GA + $\mathbf{\Phi}^{(B)}$ in the Hanoi WDN for the k-NN classifier. 140
7.3 Sensor placement results using the ES and GA in the Hanoi WDN for the Bayesian classifier. 141
7.4 Sensor placement results using GA + $\mathbf{\Phi}^{(B)}$ in the Hanoi WDN for the Bayesian classifier. 141
7.5 Results of the ES and GA methods in the Hanoi WDN for the k-NN classifier. 144
7.6 Results of the ES and GA methods in the Hanoi WDN for the Bayesian classifier. 145

7.7 Results of the proposed hybrid feature selection in the Hanoi WDN for the k-NN classifier. 146
7.8 Results of the proposed hybrid feature selection in the Hanoi WDN for the Bayesian classifier. 147
7.9 Sensor placement results using GA in the Limassol DMA network for the k-NN classifier. 151
7.10 Sensor placement results using GA in the Limassol DMA network for the Bayesian classifier. 152
7.11 Sensor placement results using GA + $\mathbf{\Phi}^{(B)}$ in the Limassol DMA network for the k-NN classifier. 152
7.12 Sensor placement results using GA + $\mathbf{\Phi}^{(B)}$ in the Limassol DMA network for the Bayesian classifier. 153
7.13 Sensor placement using only a wrapper GA method in Limassol DMA network for the k-NN classifier. 155
7.14 Sensor placement using only a wrapper GA method in Limassol DMA network for the Bayesian classifier. 155
7.15 Sensor placements of the proposed hybrid feature selection in Limassol DMA network for the k-NN classifier. 156
7.16 Sensor placements of the proposed hybrid feature selection in Limassol DMA network for the Bayesian classifier. 157

1. Introduction

This chapter introduces the topic addressed in the PhD thesis as well as the motivations to elaborate it. The structure of the PhD thesis is also presented.

1.1. Motivation

Water Distribution Networks (WDNs) are one of the most important infrastructures in cities nowadays. The concern of their efficient management has been increasing with the aim of reducing the water loss due to the need to fulfill the demand of the growing population.

This kind of infrastructure has been presenting a lower performance in practice (Kingdom et al., 2006), with an estimated loss of water, called Non-Revenue Water (NRW), around 27 % Worldwide average according to the Global Water Intelligence survey in 2008 (Global Water Intelligence, 2008).

This NRW is composed of two factors, water losses and unbilled authorized consumption (e.g., consumption through fire hydrants by firefighters). The water losses are divided into "real losses" and "apparent losses". The apparent losses are formed by errors in the measurements and measurements under-registration (e.g., consumption made by illegal connections). The real losses are the leakage in the WDN. In (Lambert, 2003), these concepts are further explained. To reduce water losses some methodologies can be applied in terms of pressure control (Mutikanga et al., 2012). On the one hand, there is the "background leakage", which is the amount of water loss due to small leaks (e.g., water loss trough pipe junctions, undetectable leaks or

small leaks that are not worth to be repaired). On the other hand, there are the leaks that are relevant and need to be repaired.

With the aim of reducing the amount of leakage in WDNs, different lines of research are open in the field of leak detection, leak isolation (leak localization) and leak estimation (quantifying the leak size). For leak isolation, two kinds of objective methodologies exists: the methods used to detect the existence of a leak in an area and the methods which allow determining the exact location. In practice, both are used in combination to find the leaks. Since the WDN are large systems, the water companies used to divide the network into District Metered Areas (DMAs), where the flow and the pressure are measured and to maintain a permanent leakage control-system: leakages in fact increase the flow and decrease the pressure measurements at the DMA entrance.

1.2. Objectives

WDNs are large scale systems which are very difficult to model with accuracy. Moreover, also some relevant parameters like the consumer demands or the leak size usually are not available, so they must be estimated or treated as uncertainty. Apart from these problems, the budget constraints limit the number of sensors that can be installed inside the WDN.

By using the model of the WDN, it is possible to generate residuals (i.e., differences between estimations provided by the model and measurements provided by sensors installed) that are indicative of leaks. Unfortunately, all the problems commented above, together with the difficult manipulation of the WDN model equations (implicit non-linear equations), limit the application of the classical Fault Detection and Isolation (FDI) structured residual framework, where it is assumed that there is a clear binary relation between faults (leaks) and residuals. In practice, all the leaks affect all the residuals to some extent and this complicates the analysis of the residuals. In this context, the techniques developed in the machine learning field are potentially useful.

The objective of this PhD thesis is to research new techniques to locate the leaks in WDNs using a model-based approach and machine learning techniques with the aim of achieving accuracy and robustness against the associated uncertainties (not precisely known nodal demands, unknown leak size, noise in the measurements, WDN parameter uncertainties, etc.). Moreover, a data-driven approach where the model is not needed is studied. Additionally, the problem of optimal sensor location, in accordance with the proposed leak localization methods, is also studied. Finally, a leak detection technique is developed to enhance the leak localization approach while reducing the uncertainties and maximizing the use of the available data in the diagnosis.

1.3. Outline of the PhD Thesis

This PhD thesis has been organized in seven chapters. Chapter 2 presents the water distribution network model, the state of the art regarding to the leak detection and localization tasks and the introduction of the machine learning techniques used in the following chapters. Then, the case studies where the proposed methods are tested are introduced in Chapter 3. Chapter 4 presents a leak detection method. Chapter 5 presents a model-based leak localization approach while in Chapter 6 a data-driven leak localization approach is proposed. The problem of sensor placement for the proposed leak localization approaches is addressed in Chapter 7. Finally, Chapter 8 draws the conclusions, the contributions of this PhD thesis and the future work.

1.3.1. Chapter 2: Background

Chapter 2 presents the theory of the modeling of WDNs and how the models are used to generate data for the proposed methods presented later. The past and current leak detection and localization approaches in the literature are revised as well as the sensor placement techniques available for them. Finally, a brief introduction to the machine learning terminology and the techniques used in the proposed leak

detection and localization methods are theoretically described.

1.3.2. Chapter 3: Case Studies

In Chapter 3, the different case studies where the proposed methods have been applied, including some real cases, are described.

1.3.3. Chapter 4: Leak Detection

Chapter 4 presents a leak detection technique based on a data-driven method that makes use of the self-similarity and an improved intersection-of-confidence-interval change detection test to work specifically with the detection of leaks, which is able to estimate when the leak has started and to estimate the leak size.

Related publications:

Soldevila, A., Boracchi, G., Roveri, M., Tornil-Sin, S., & Puig, V. (2018). Methodology for Leak Detection and localization in Water Distribution Networks. *To be submitted at the IEEE Transactions on Systems, Man and Cybernetics: Systems.*

1.3.4. Chapter 5: Model-Based Leak Localization

Chapter 5 presents a model-based leak localization methodology that combines the use of hydraulic models with machine learning techniques, particularly the non-parametric k-NN algorithm and the parametric multi-class Bayesian classifier, to deal with the uncertainties that exist in practice and, therefore, improve robustness.

Related publications:

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

Soldevila, A., Fernandez-Canti, R. M., Blesa, J., Tornil-Sin, S., & Puig, V. (2016,

June). Leak localization in water distribution networks using model-based Bayesian reasoning. In *European Control Conference (ECC), 2016* (pp. 1758-1763). IEEE.

Soldevila, A., Blesa, J., Tornil-Sin, S., Duviella, E., Fernandez-Canti, R. M., & Puig, V. (2016). Leak localization in water distribution networks using a mixed model-based/data-driven approach. *Control Engineering Practice, 55*, 162-173.

Soldevila, A., Fernandez-Canti, R. M., Blesa, J., Tornil-Sin, S., & Puig, V. (2017). Leak localization in water distribution networks using Bayesian classifiers. *Journal of Process Control, 55*, 1-9.

Soldevila, A., Tornil-Sin, S., Blesa, J., Fernandez-Canti, R. M., & Puig, V. (2017). Leak Localization in Water Distribution Networks Using Pressure Models and Classifiers. In *Modeling and Monitoring of Pipelines and Networks* (pp. 191-212). Springer International Publishing.

1.3.5. Chapter 6: Data-Driven Leak Localization

Chapter 6 presents a new methodology for leak localization without the use of hydraulic models. This is done by means of an interpolation technique and the Bayes rule to infer from the new measurements information that is present when leaks are in a particular location inside the network.

Related publications:

Soldevila, A., Blesa, J., Fernandez-Canti, R. M., Tornil-Sin, S., & Puig, V. (2018). Data-Driven Approach for Leak Localization in Water Distribution Networks. *Submitted to the Water Resources Management.*

1.3.6. Chapter 7: Sensor Placement

Chapter 7 presents different sensor placement methods for the different leak localization approaches presented in the two previous chapters. The use of a genetic algorithm and feature selection techniques are used standalone and in combination with other approaches to retrieve suitable sensor placement to enhance the leak localization performance.

Related publications:

Soldevila, A., Tornil-Sin, S., Fernandez-Canti, R. M., Blesa, J., & Puig, V. (2016, September). Optimal sensor placement for classifier-based leak localization in drinking water networks. In *3rd IEEE Conference on Control and Fault-Tolerant Systems (SysTol)*, 2016 (pp. 325-330).

Soldevila, A., Blesa, J., Tornil-Sin, S., Fernandez-Canti, R. M., & Puig, V. (2017). Sensor Placement for Classifier-Based Leak Localization in Water Distribution Networks. In *Modeling and Monitoring of Pipelines and Networks* (pp. 213-233). Springer International Publishing.

Soldevila, A., Blesa, J., Tornil-Sin, S., Fernandez-Canti, R. M., & Puig, V. (2018). Sensor placement for classifier-based leak localization in water distribution networks using hybrid feature selection. Computers & Chemical Engineering, 108, 152–162.

Soldevila, A., Blesa, J., Fernandez-Canti, R. M., Tornil-Sin, S., & Puig, V. (2018). Data-Driven Approach for Leak Localization in Water Distribution Networks. *Submitted to Water Resources Management.*

1.3.7. Chapter 8: Conclusions

Chapter 8 presents the conclusions of this PhD thesis, and highlights the main contributions made during its elaboration. Finally, some future work to be done in line of this PhD thesis is proposed.

2. Background

This chapter introduces the concept of WDNs and their modeling. The past and recent leak detection and localization techniques are presented. Finally, the basic terminology of machine learning and the methods of that field used in this PhD thesis are explained.

2.1. Water Distribution Networks

WDNs, usually organized in DMAs, are large scale systems formed by inlets (that typically correspond to reservoirs) that feed the network with water, pipes that distribute the water across the network and nodes which can be junctions between pipes or points where the consumer users are connected with the network. An example of a DMA network is depicted in Figure 2.1. Reservoirs are placed in an elevated place to assure a good pressure service for the costumer where Pressure Reducing Valves (PRVs) can be used to regulate pressure with the aim of reducing the background leakage and extent the life of the network.

WDNs can be modeled with the non-differential Hazen-Williams equation as a static system considering that the changes in demands and flows are slow enough to consider the system operating in steady-state. The Hazen-Williams equation describes the flow in the pipes by

$$fl_{i,j} = \left(\frac{h_i - h_j}{\mathfrak{r}} \right)^{a-1} \tag{2.1}$$

where $fl_{i,j}$ is the flow in the pipe between the nodes i and j in [m^3/s] which is positive

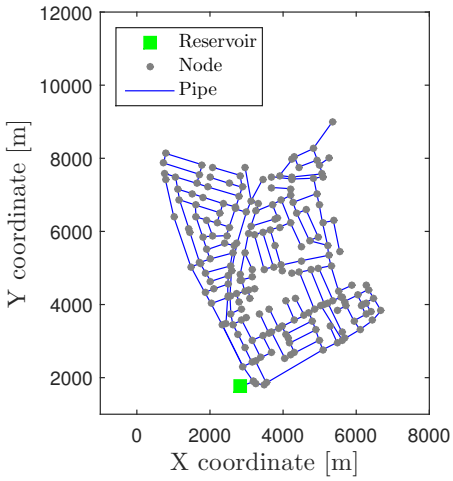


Figure 2.1: Limassol topological DMA network.

from i to j, h_i and h_j are the head (pressure taking into account the elevation of the node) of nodes i and j respectively in [m], $\mathfrak{r}$ is a adimensional coefficient that is depends on the physical characteristics of the pipe

$$\mathfrak{r} = 1.2216 \cdot 10^{10} \frac{L}{C^a D^{4.87}} \tag{2.2}$$

where L is the length of the pipe in [m], C is the roughness of the pipe, D is the diameter of the pipe in [mm], a is the flow exponent coefficient, which is equal to 1.852 and the values .

Additionally to (2.1), the flow balance can be established in the nodes by the energy conservation law, which is described as $1.2216 \cdot 10^{10}$ and 4.87 are the values are characteristic of the function.

$$f_i - d_i = 0 \tag{2.3}$$

where d_i is the demand at node i and f_i is the sum of the flows that pass through the node i, both in [m^3/s]. The signs of the flow at every pipe are defined in (2.1). The demands of every node can be calculated using the total inflow water and a pattern distribution. Usually the demands at each node are estimated using the billing records. These records are used to calculate the daily average consumption

of each node with respect to the global consumption that is shaped with a low time scale pattern distribution of the WDN water consumption. Then, the demands in the nodes are considered that have this fixed pattern distribution (i.e., every node always has the same proportion of the total water inflow into the WDN), then the demand in every node is computed as

$$\hat{d}_i = \alpha_i \tilde{d}_{WDN} \tag{2.4}$$

where $\hat{d}_i$ is the estimated demand at the node i in [m^3/s], α_i is the normalized proportional outflow in node i and $\tilde{d}_{WDN}$ is the total inflow water in the WDN in [m^3/s], which is a measured value.
Deriving from (2.4) the following condition is fulfilled

$$\sum_{i=1}^{n_n} \alpha_i = 1 \tag{2.5}$$

where n_n is the total number of nodes in the WDN.
Considering (2.1), (2.4) and (2.3), the knowledge of the demand pattern distribution and the hydraulic characteristics (i.e., pipe shape, length, etc.), and the measurement of the boundary conditions $\tilde{\mathbf{c}}$ (Pressure Reducing Valves (PRVs), total water consumption, etc.), the WDN hydraulic system can be simulated by means of a hydraulic simulator that provides a numerical solution using as e.g., the Newton-Raphson method. In this PhD thesis, the hydraulic simulator Epanet 2 (Rossman, 2000) is used.

2.1.1. Modeling the Leak as an Extra Demand

As it will be discussed later, it is a common approach to consider that leaks can only occur at nodes (note that a virtual node placed in pipes can be created with a demand equal to zero). Thus, the leak can be modeled as an extra demand. This leads to a new pattern distribution to accommodate that new leak considered as an extra demand.

To adjust the measured total water consumption $\tilde{d}_{WDN}$ with the leak and the estimated demand pattern distribution, a new demand pattern distribution where the leak amount is removed from all nodes (except from those that the consumption is already zero) in a proportional way according to their average consumption level, and the leak is added to the node where it is simulated. The new demand pattern distribution is calculated as

$$\alpha_i^{(l_j)} = \begin{cases} \alpha_i - \alpha_i \frac{l_j}{\tilde{d}_{WDN}} & i \neq j \\ \alpha_i - \alpha_i \frac{l_j}{\tilde{d}_{WDN}} + \frac{l_j}{\tilde{d}_{WDN}} & i = j \end{cases} \tag{2.6}$$

where l_j is the leak (with leak size l in [l/s]) at the node j and $\boldsymbol{\alpha}^{(l_j)}$ is the new demand pattern distribution with a simulated leak l at node j.

2.1.2. Uncertainty Modeling Using Artificial Data

With the aim of developing robust methods, demand and leak uncertainties are generated and artificial noise is added to the measurements. These uncertainties are artificially generated using the methodology presented in (Cugueró-Escofet et al., 2015b), which is described in the following. First, the noise in the measurements is added to the pressure values as

$$\bar{p}_i = p_i + \bar{\nu} \tag{2.7}$$

where

$$\bar{\nu} = \nu^{(u)} rand \tag{2.8}$$

and $\bar{p}_i$ is the generated pressure at the node i with noise in [m], $\nu^{(u)}$ is the amplitude of the noise and $rand$ is a random value in the range [-0.5,0.5]. For the demand uncertainty, the demand pattern distribution is modified as

$$\bar{\alpha}_i = \alpha_i + \frac{\alpha_i \alpha^{(u)} rand}{100} \tag{2.9}$$

where $\bar{\alpha}_i$ is the demand in the node i with uncertainty in [m] and $\alpha^{(u)}$ is the amplitude of the demand uncertainty in [%] and *rand* is a random value with Gaussian distribution in the range [-0.5,0.5]. But, to accomplish that all the normalized demands satisfies (2.5), a normalization is performed

$$\bar{\alpha}_i = \frac{\bar{\alpha}_i}{\sum_{j=1}^{n_n} \alpha_j} \tag{2.10}$$

Then, the new generated demands with uncertainty can be computed as

$$\bar{d}_i = \bar{\alpha}_i \tilde{d}_{WDN} \tag{2.11}$$

where $\bar{d}_i$ is the demand at node i considering the uncertainty. Finally, to generate the leak size uncertainty, the following equation is used

$$\bar{l}_i = l_i + \frac{l_i l^{(u)} rand}{100} \tag{2.12}$$

where $\bar{l}_i$ is the leak in the node i with uncertainty in [l/s], $l^{(u)}$ is the amplitude of the leak uncertainty in [%] and *rand* is a random value with Gaussian distribution in the range [-0.5,0.5].

Moreover, to have more realistic data sets, sequences of daily global demands are artificially generated. The demand pattern distribution is usually considered fixed in time (Wu et al., 2011), and only changes the magnitude of the total WDN consumption, that is commonly measured in practice and presents a daily pattern. To create a daily pattern in simulation a filtered real data pattern from another WDN is used then a Gaussian noise is added in this pattern to create daily patterns as follows

$$\bar{d}_{WDN} = \hat{d}_{WDN} + d_{WDN}^{(u)} rand \tag{2.13}$$

where $\bar{d}_{WDN}$ is the total consumption instantaneously with uncertainty in [l/s], $\hat{d}_{WDN}$ is the estimated total consumption obtained from historic records in [l/s], $d_{WDN}^{(u)}$ is the amplitude of the uncertainty in [l/s] and is a random value with Gaus-

sian distribution in the range [-0.5,0.5].

2.1.3. Uncertainty Modeling with Real Data

When real data is available, another approach can be used to generate a more accurate model of the uncertainties that uses this real data in combination with the hydraulic model, where the demand uncertainty, the noise in the measurements and the modeling error are taken into account. In this case, the discrepancy between the real measurements available and the estimated ones generated with a hydraulic simulator under the same conditions (i.e., measured boundary conditions $\tilde{\mathbf{c}}$, measured global consumption $\tilde{d}_{WDN}$ and estimated nodal demands $\hat{\mathbf{d}}$) is used. So, for the pressure sensors can be described as

$$\mathfrak{a}_i = \tilde{p}_i - \hat{p}_i \tag{2.14}$$

where $\mathfrak{a}_i$ is the discrepancy between the estimated values and the real measurements in [m].

Once (2.14) is applied to all the data, the result is a cloud of points supposed to be around zero where the mismatches corresponds to the uncertainty of the hydraulic model, the noise of the measurements and the nodal demand uncertainty. Usually, these mismatches are not around zero due to some offset of the values provided by the difference of the sensors and the hydraulic simulator. To deal with this, the $\mathfrak{a}_i$ values previously computed are corrected using the average value of the discrepancy as

$$\mathfrak{a}_i^{(c)} = \mathfrak{a}_i - \text{mean}(\mathfrak{A}_i) \tag{2.15}$$

where $\mathfrak{a}_i^{(c)}$ is the centered (offset removed) discrepancy and $\mathfrak{A}_i$ is the vector of discrepancies for the measurement i.

Then, the hydraulic model is used to generate the sensitivity matrix, where the offsets between the estimated and the real measurements are removed using the

average discrepancy of each measurement. After that, the discrepancy $\mathfrak{a}_i^{(c)}$ is added to the values of the sensitivity matrix (i.e., centered from zero to each leak value in the residual space). In this way, each leak fault simulated without uncertainty, which is a point in the residual space, now has a cloud of points around them that represents the uncertainties of model, noise and nodal demand.

This kind of approach is useful when the network is really large and time-consuming for obtaining these data allowing to catch the model uncertainty. But, this approach has the drawback of not representing as good as the other approaches the particular nodal demand uncertainty of each node.

2.2. Leak Detection and Localization

In the field of leak detection, estimation and isolation in WDNs, several techniques are applied and more are in development. In (Fanner et al., 2007; Puust et al., 2010; Mutikanga et al., 2012), recent reviews are presented about the most widely used methods and techniques presented so far. Another interesting review is presented in (Li et al., 2015), where the methods are classified as hardware based and software based. In (Wu and Liu, 2017), data-driven approaches, focused mainly in the leak detection problem, are reviewed. In this section, the current and past techniques related with this PhD thesis are revised.

These techniques are grouped into different types according their nature, but some of them can be classified into more than one group. The selection of one or another group is made here according with the emphasis made by the author of the original work into the corresponding field.

Due to the different approaches to the considered problem, different areas of application are involved. Some of them are applicable to the entire WDN, others are applicable to DMAs and finally, some of them applicable to single pipes.

2.2.1. Leak Assessment

To know the efficiency of a WDN there are three generalized methods to estimate the total real losses of water (Puust et al., 2010). The Top-down method uses the inflow water into the WDN and removes the outflow water registered and an estimation of the apparent losses leaving the system. The remaining amount of water is the estimated real losses. On the other hand, the Bottom-up method (Mazzolani et al., 2015) does the same but using the Minimum Night Flow (MNF) which is the period of the day when the customers demands are minimum and the leakage is higher (in percentage), and then the result is extrapolated to the diurnal time. Finally, the method based on Principal Component Analysis (PCA) that uses all the data available (as, e.g., water reports, annual flows and patterns, state of the WDN) to find the principal indicators to get the most accurate estimation. Usually, all of the three methods are used in combination to get the best possible performance.
Other approaches (Roma et al., 2015) use the WDN calibration to assess the amount of background leakage in a network using the discrepancies between the real network and the a hydraulic model of the system. In (Almandoz et al., 2005), the hydraulic model is used along with the historical data of the flow at the inlet. In this method, the real losses are considered since they are only ones that depend on the pressure allowing the separation from the apparent losses and quantify the different water losses.

2.2.2. Step Testing

With this technique, the WDN is divided into several areas by means of valves, and by closing the water in some areas, usually at night, the leak can be found in the area with the abnormal water consumption. The problem with the application of this technique is that the WDN needs to be built with the idea of applying this technique or otherwise it will be difficult to be used successfully. Another drawback of this approach is that some parts of the WDN will experiment water cuts (with

the corresponding cut of the service for the clients) during the test.

2.2.3. Acoustic and Vibration Techniques

A large variety of techniques using the sound (or the vibration) generated by the friction between pressure water waves and pipe walls and the particular sound (and vibration frequencies) that appears when a leak exists are developed and used with success in the field. This type of methods are commonly used to find the exact location of the leak once another method had pointed the area in which the leak is. A common drawback is the different performance of the method related to the pipe materials, for example in (Zhang and Guo, 2011) a study of the acoustic emissions in cast iron pipes is presented.

Acoustic Logging (AL)

This technique uses hydrophones placed all over the WDN to detect the particular sound emitted by leaks to localize the leak in the area where the hydrophone detected the suspicious sound. The drawbacks of this technique (and also for all the acoustic techniques) is that the noise provided by external factors (e.g., generated by traffic) can deeply affect the measurement. Also, it should be noted that in this case the leak is not exactly localized but the area where it exists is reduced.

Leak Noise and Vibration Correlators (LNC)

In this technique, the sound generated by the leak is registered by two (at least) microphones (or hydrophones) attached to pipe stems. The sound can be replaced by vibrations, and detected using accelerometers (Khulief et al., 2012; Martini et al., 2015; Zhong et al., 2015). Then, the specific sound of the leak is detected and the time delay to reach all the microphones is calculated by correlating the specific sound pattern caused by the leak. The leak can be located knowing the time delay, the distance between the microphones and the speed of the sound in the pipe by means of triangulation. The drawbacks of this approach are that every pipe needs to be

tested until the leak is found. This can be a very time-consuming task. Moreover, the performance of this technique changes with the pipe materials (Gao et al., 2009), and the distance between sensors is required with precision although there exist some techniques to avoid the necessity of this information (Yang et al., 2008). Since the surrounding environment can effect the measurements due some undesired added sound that can difficult the task, some efforts are made to reduce their impact (Ionel et al., 2010).

Pig-Mounted Acoustic (PGM)

In this technique, a device is inserted into the WDN which contains a microphone to detect the leak. Then, from outside the WDN, with the recorded sound along with the position in the WDN where the device has been moved, the leak can be located and the leak size estimated by the intensity of the sound (Mergelas and Henrich, 2005). This technique is only applicable to pipe mains.
Similar approach is presented in (Adhikari, 2014) where a listening device is carried over (at the surface) the network, where the acoustic measurements are analyzed using Fast Fourier Transform (FFT) to identify frequency peaks emitted by leaks. In this case, this technique is applicable only in networks (or portion of networks) where the device can record the sound emitted by the leak.

2.2.4. Surface Analyzer Methods

Different approaches use different characteristics that change when leak appear and that can be monitored from the surface of the network by carrying an apparatus over the network to analyze those particular characteristics.

Tracer Gas Technique (TGT)

The TGT is one of the techniques with better performance. This method uses a gas with the particularity that it is lighter than air. This gas is injected into the WDN and it escapes through the leaks existing in the WDN, even the smallest ones, going

to the surface. Then, with a gas detector on the surface the leaks can be located exactly (Hunaidi et al., 2000). The drawbacks are that is very expensive to fill the entire WDN, and its time-consuming to carry the detector all over the network until the leak is found. On the other hand, it has the benefit that not qualified personnel is required.

Ground Penetrating Radar (GPR)

This technique uses an emitter of electromagnetic waves that are propagated through the ground and scattered back to an electromagnetic detector. The different electromagnetic properties in the ground are shown with the detector, and one of the reasons that can produce a change in these properties is the water in the ground escaped through the leaks (Hunaidi and Giamou, 1998; Stampolidis et al., 2003; Farley and Hamilton, 2008). This technique performs well with all the pipe materials, being non-intrusive and not expensive. However, it requires an expert operator to analyze the image produced by the detector. And even with this, it is easy to obtain false positives (e.g., junctions or valves that change the image). Thus, it can be difficult to use it in all places (e.g., a pipe which passes under a highway) and the need of moving across all the network makes this technique very time-consuming.

Thermography

When leaks occur and the ground surrounding the pipe is affected by the water leaked, the temperature of the ground can change. Thus, by using an infrared camera, the area affected by the water loss through the leak can be seen as a cooler or warmer area through the camera (Hunaidi et al., 2000; Fahmy and Moselhi, 2010; Shakmak and Al-Habaibeh, 2015). This technique has the same drawbacks as the GPR technique.

2.2.5. Hydraulic Analysis

Other kinds of techniques use the hydraulic properties of the WDNs, such as the pressure or the flow rate, and several parameters of the hydraulic model.

The hydraulic model of WDNs is highly studied (Wu et al., 2011) but usually the model obtained is not accurate enough (e.g., the roughness of pipes changes with use). So a model calibration is needed to obtain adjusted results with the real WDN behavior (Walski et al., 2003; Savić et al., 2009; Ostfeld et al., 2012). Even with the calibration, the hydraulic model simulations can not reproduce perfectly the WDN behavior. Furthermore, the problem of numerical errors added in the simulation process can appear.

The hydraulic analysis can be done in steady state or in transient. In both cases, there are benefits and drawbacks. Usually when the WDN has low pressures and higher flows the transient analysis could work better. On the contrary, the steady state analysis works better with high pressures and lower flow rates (Ferrante et al., 2014).

Steady State Analysis

These methods use the hydraulic model in steady state (with the assumption that the WDN stays in steady state in a short period of time) to obtain and work with the hydraulic information. It is usually assumed that the leak must occur only in nodes.

Inverse Analysis (IA)

The techniques based on IA use the hydraulic model equations which describe the WDN but with unknown added parameters to be estimated, which are the diameter of the leak hole in every node along with the parameter of roughness coefficients of every pipe in the network. With the measurements available from the WDN, the estimation of those parameters is done, usually with the objective of minimizing the difference between the measurements and the predicted ones by the hydraulic model (i.e., pressure and flow rate) (Pudar and Liggett, 1992; Sala and Ko, 2014).

This technique has several drawbacks: the hydraulic model (as e.g., pipe parameters and nodal demands) must be accurate, and the leak hole can have several types of shapes which in practice correspond to different coefficients in the standard leak model since the shape of the hole is not known, and its behavior changes depending on the pipe material (Greyvenstein and van Zyl, 2007).

In (Puust et al., 2006), the problem is solved using the Shuffled Complex Evolution Metropolis (SCEM-UA) algorithm that poses the problem of leak hole estimation (the pipe roughness coefficients in this case are not estimated) as probability to be maximized. More recently, the problem is solved using simulated annealing (Sousa et al., 2015), where also a sensor placement based on graph theory is proposed.

Sensitivity Analysis

The use of sensitive analysis has been used in large variety of techniques usually combined with pressure and flow measurements.

One of these techniques is based on the use of the residuals, which are the differences between real and expected values obtained by simulation using the hydraulic model in steady state. Usually a sensitivity matrix with information about the leak signature of every leak case is obtained using the hydraulic model (that is, by simulating the WDN with a leak in every node). Then, the residual obtained with the real measurements is compared with the information stored in this matrix.

Several approaches have been developed using this approach. In (Pérez et al., 2010; Pérez et al., 2011), the matrix is binarized using a threshold that allows obtaining a binary leak signature for each (where the complete set of signatures is called Fault Signature Matrix (FSM)). Analogously, the residual obtained is compared with the columns in the matrix that contains the signature of each node, the most similar (using some metric as the Hamming distance) signature is where the leak is assumed to be located. In (Quevedo et al., 2012), the matrix is not binarized and the node candidate is the one that has the most correlated signature with the residual. This work was extended by taking into account the demand uncertainty in (Pérez et al., 2016). In (Casillas et al., 2012), the angle between every leak signature in the sensitivity matrix and the residual is calculated, and the closest one (minimum

angle) is the node candidate. Additionally, a time horizon is introduced into the analysis. An improved technique (Casillas et al., 2015a) to reduce the impact of the non-linearity leak behavior one residual is used to normalize the others and then, by means of the Euclidean distance, the leak is located finding the minimum distance between the residual obtained and the different columns of the sensitivity matrix. An alternative to the use of the sensitivity matrix approach, it is based on the application of the structural analysis as explained in (Rosich et al., 2014).

In (Blesa et al., 2012), the demand uncertainty and noise are modeled as zonotopes in the residual space using Linear Parameter Varying (LPV) parity equations. Then if the residual obtained is out of the zonotpes, the network has a leak. For the localization task, the residuals are binarized and compared against to the FSM in a time horizon.

In (Sanz et al., 2015), the profile demands are calibrated using a discretitzation of the network (according to the sensors placement of the network), and then the water demand profile of every node is calibrated according to their basic demand estimation and adding a function of the components of every discretized zone (the weight of every demand component is determined by their geographical location). For the detection, the demands obtained are compared against a threshold, in order to check if the demand is large enough with respect to the normal consumption to consider the presence of a leak. Two different leak localization techniques are proposed, the discretized zone with the biggest increment of demand is one, and the other one uses Pearson correlation to search the node with the most different behavior compared with the nominal.

In (Verde, 2005), a multi-leak detection and localization approach is presented, which is able to detect and locate up to two leaks in a pipe main using flow and pressure sensors at the end of the pipe. The technique divides the pipe in three subsystems and try to identify (by estimating parameters using the measurements) the positions where the system (i.e., the leak locations are the limits of each subsystem) must be splitted. This is only applicable to pipe mains.

In (Escalera et al., 2012), a multi-leak scheme for leak detection and localization is

used, where the residual sensitivity matrix (obtained using a hydraulic simulator) is transformed using Complex Shannon wavelets and then is demodulated using phase-quadrant demodulation to extract its information phase, which is able to separate the different phases for each leak in the network. Then, the node with the residual in the sensitivity matrix with the most similar phase with the residual being analyzed is the leak node candidate.

In (Narayanan et al., 2014b), only the flow measurement in the inlet is used to detect, localize and estimate the leak size. For the detection, the current flow is compared against historical records. If the discrepancy is high in a time window, then is considered that the leak exists, and the amount of discrepancy is considered as the leak size. For the localization, a simplified hydraulic model is created without calibration, where the connected nodes by pipes with low flow resistance are grouped to simplify the number of nodes of the simplified network. Finally, the amount of flow is used to determine which are the groups of nodes that can produce that amount of leak. The result is improved with a time horizon giving the groups of nodes in which are possible that the leak is placed over the time.

In (Ishido and Takahashi, 2014), a new indicator is presented called Head Loss Ratio (HLR), which is obtained combining the heads (pressure taking to account the elevation) measurements. This indicator is more sensitive to leaks than the raw measurements. The drawback of this indicator is that its only applicable in networks with specific characteristics (only one reservoir, no pumps, etc.).

In (Meseguer et al., 2014), an hybrid approach is presented where the sensitivity approach is combined with the IA approach to deal with the multi-leak case. In this method, the FSM is generated in two ways, on the one hand using the real measurements and the hydraulic simulator without artificial leaks, and the other hand by using the IA and the hydraulic model without leaks. The aim o the difference of these two FSM, by using a genetic algorithm to solve the identification problem. This method provides the leak or leaks localization along with their estimated size.

In (Bakker et al., 2014), a simplified hydraulic model (uses simpler equations to describe the relation between the inlet flow and pressure and the pressure inside

the network) is created using past data from the same points characterized with the model. Then, the output of the model is compared with the current measurements to generate residuals. The last residuals (moving window) are compared against a threshold to detect a leak.

In (Jensen and Kallesøe, 2016), a reduced model of the network is proposed where the sensors are placed such that the leaks only can affect one pressure sensor, and at the same time, the sensors sensitivity must cover all the network. Then, for the detection each pressure measurement is contrasted against the prediction of the reduced model. If the residual value exceed a threshold, then the leak is detected, and from the zone (nodes) that cover the sensitivity of that sensor the node candidates are proposed.

Transient Analysis

One of the most active areas of research in the leak monitoring is the transient analysis. A review of this approach is presented in (Colombo et al., 2009). This family of methods uses the transient information generated by an event (e.g., a change in a valve) to collect information to infer the diagnosis. The advantage compared with the steady state methods is the amount of information obtained in a short period of time. In general, transient analysis is used only in pipe segments or small networks.

Some of these approaches have the drawback that it is necessary (in most of the available techniques) to create a repetitive pressure wave (to have the same input to the system that allows to accurately analyze the output) in the WDN capable of showing the specific characteristics needed to analyze specific parameters. This can be difficult to do and dangerous for the WDNs infrastructure health.

Inverse Transient Analysis (ITA)

This technique, introduced for the first time in (Liggett and Chen, 1994), is the same as the one in steady state approach but using a transient model instead. The drawbacks of this technique are the same than the steady state method. In this case, the identification requires a lot of time to achieve an optimal solution. Some research

efforts in this area are focused on developing suboptimal solutions (Vítkovský et al., 2000), and improving and extending the hydraulic model parameters (Covas et al., 2004; Wu and Sage, 2006; Covas and Ramos, 2010).

In (Srirangarajan et al., 2013), the pressure signals are analyzed by a multi-scale wavelet analysis to remove noise and decompose the original signal into four levels to be analyzed, more precisely, the coefficients of the levels three and four are analyzed by the CUmulative SUMmation (CUSUM) for detection purposes. Once a leak detection is raised, the estimated leak starting time can be obtained searching for the moment that the peak in the signal that provoked the alarm started raising. Finally, a leak localization is proposed by triangulation using the difference in the estimated leak starting time into each sensor.

In (Delgado-Aguiñaga et al., 2016), a multi-leak (the leaks must occur at different times) detection and localization method is proposed in pipe mains where sensors of flow and pressure are installed at the ends of the main. Then, the dynamic model of the system is used to identify the leak (by the mismatch between the current measurements and the model prediction). Once the leak is detected, a Kalman Filter (KF) is used to update the states of the system and incorporate the leak. Then, a new leak can be detected using the updated system.

Leak Reflection Method (LRM)

When a wave (artificially generated) arrives at a leak, part of this wave is reflected and then knowing the wave velocity and the difference in time between the original wave and the reflection generated by the impact with the leak hole permits to know the localization of the leak. The relationship between the two wave magnitudes (the original and the reflected) can be used to estimate the leak size (Beck et al., 2005; Lee et al., 2007; Soares et al., 2012). This technique is only applicable in pipe segments and has the drawback that it is very important to know with precision the scenario without leak, because other reflections can be produced by other factors in the pipe (as e.g., pipe junctions) and can be wrongly identified as leaks.

In (Nguyen et al., 2018), a special methodology for the generation of the transient events is proposed using pseudo random binary sequences to minimize the correlation

of the different events than leaks with the aim to increase the signal to noise ratio, and therefore, improve the localization.

Impulse Response Analysis (IRA)

IRA (Ferrante and Brunone, 2003a,b; Misiunas et al., 2004; Ferrante et al., 2007) uses the continuous monitoring (high sampling frequency) state of the WDN using at least two pressure sensors with the aim of detecting an abrupt change in the pressurized state (i.e., transient effect produced by a Negative Pressure Wave (NPW), which is due to a sudden pipe burst). Then, the leak can be located using the transient signal obtained (the time arrival to the sensors and the pressure variation to perform triangulation). The benefit of this technique is that it is not necessary to produce an artificial transient event, but has the drawback that it is necessary to monitor the WDN continuously which can be difficult with the standard sensor placement methodologies (i.e., the sensors usually work using batteries and consequently have limited autonomy).

In (Zan et al., 2014), a Joint Time Frequency Analysis (JTFA) is proposed, where the pressure measurements are sampled at a high frequency. Then, a one-dimensional wavelet is used to filter high frequency noise, then a Short-Time Fourier Transform (STFT) is applied to obtain the spectrogram of the signal. A Gabor Transform is used to remove the unnecessary part of the spectrogram and the remaining part is processed by a moving average function. Finally, the obtained indicator is compared against a threshold (obtained from historical data without leak). For the localization of the leak, an attenuation function between the leak position and the sensors position is used to estimate the expected distance values for each potential leak location. Then, the nodes are ranked using the Johnson's algorithm by minimizing the difference between the expected and the computed distances values. The node on top of the rank is the node candidate.

More recently, in (Lee et al., 2016), a variant of the previous method is presented where the measurements are denoised and decomposed to different levels using a wavelet transform. Then, the CUSUM technique is applied in these levels to detect the negative wave of pressure, the NPW, to detect the leak. For the localization,

the difference in the time that this negative wave is detected by the sensors is used to triangulate the leak position.

Frequency Response Method (FRM)

This technique uses induced transient events (e.g., a predefined change in the state of a valve) and the information obtained from these events are analyzed in the frequency domain. When a leak exists in the WDN, particular frequencies appear in the frequency domain that do not exist in the leak free scenario. Considering this fact, the leak can be detected in the WDN. The information of localization and size can be also obtained from the frequency domain. However, it is a difficult task to localize the leak and even more to estimate the leak size (Mpesha et al., 2001). Thus, it is only realistic to apply this technique in small networks or pipe segments. Other approaches are based on the analysis in the frequency domain of the transient damping effect (the damp of a pressure wave is produced by the effect of friction with pipe walls, and when leaks exist this damp effect is augmented). Then, using the difference between the state of the harmonics in a leak-free scenario and in the leak scenario, the leak can be detected, located and the leak (hole) size estimated (Wang et al., 2002). However, this technique is only applicable to pipe segments or small networks. In (Nixon et al., 2006), the range of validity of the method is studied.

In (Covas et al., 2005), a Standing Wave Difference Method (SWDM) is used and consists on generating a steady oscillatory flow with the particularity that the frequency oscillation is chosen based on the leak resonant frequency (i.e., the time of wave pressure to reach the leak and return is an odd multiple of the half wavelength of the excitation frequency). Then, the frequency domain is analyzed to find the difference in the harmonics to detect and localize the leak. This method only works when the wave generator and the pressure sensor are close to the leak.

2.2.6. Analysis Using Statistical Methods

The analysis of different measurements of the WDNs and their comparison against historical records or the evaluation of signals through statistical methods are used to detect leaks, estimate their size, estimate the time that leak has appeared and assess where the leak is located.

In (Buchberger and Nadimpalli, 2004), the historic data of the flow measurements at the inlet is analyzed where the biggest value of the hourly mean and standard deviation is searched. Then, it is checked if the new measurements exceed those two indicators and if this is the case in one of them the leak is detected. The leak size is estimated when the leak is detected by computing the difference in both means.

In (Misiunas et al., 2006), a leak detection is presented using the measured flow rate at the inlet and the CUSUM technique. Once a detection is raised, the difference with the nominal flow rate is used to estimate the leak size, and the moment that the CUSUM feature started to raise until the detection is made is used to estimate the leak starting time. This leak starting time is used then, along with the pressure measurements inside the network, to extract the difference in the measurements with the values expected with a normal behavior. These differences in the pressure values are then compared with the ones obtained through simulations where the estimated leak size is added at every node. The case of a node with the artificial leak that provides the most similar output to the measured ones, is the node candidate.

In (Ye and Fenner, 2011), a Kalman Filter (KF) is used to generate residuals (using flow or pressure measurements at the inlet), if the residual exceeds a threshold, then a leak is detected. This method is able to detect new leaks even if there are already other leaks in the network. Also, this work concludes that flow is a better variable to detect leaks than the pressure.

In (Eliades and Polycarpou, 2012), the historical flow measurements at the inlet is used to construct Fourier series for prediction purposes. Then, the actual measurements are used to generate residuals to be analyzed by the CUSUM technique. Finally, the features are compared with a threshold for the leak detection purposes.

The leak size estimation is done by comparing the actual flow consumption and the Fourier series prediction.

In (Fusco and Ba, 2012), the nodal demands are estimated using the model and the measurements available (inverse analysis in steady state), and the Z-test is performed over those estimations using the historical demand values and a threshold is used to decide if a demand in a node is large enough to indicate that is due to a leak.

The detection of leaks using the inlet flow measurements and a KF to predict the measurements to compute residuals, which are matched against a threshold obtained using the mean and the standard deviation of the measurements, is presented in (Ye and Fenner, 2014a).

In (van Thienen, 2013), a method to compare the current inlet flow measurements with the historical ones, called Comparison of Flow Pattern Distribution (CFPD), is used to create a graph with the aim of facilitating the decision for an expert (although it is not an automated method) the detection of a leak and their size estimation.

In (Romano et al., 2012), an abnormal (leak among other events) detection technique is presented. First, the data is preprocessed selecting only some daily patterns (three statistical tests are used in parallel, all using the daily mean and standard deviations and user-defined thresholds, and the remaining patterns are stored in a data set called NOP (Normal Operating Patterns)). Then, an Artificial Neural Network (ANN) is created using the NOP data set for training and testing. From the discrepancies between the training and testing data, the mean and the standard deviation is computed for evaluating a statistical test. The outputs of the three indicators feed to two inference systems (both Bayesian Networks (BNs)). One is used to assess the probability that something has occurred using the outputs and the daily average difference between the stored values and the current ones (which can also be used to estimate the leak size). The other one is used to raise an alarm (event detection) by taking into account the three outputs of the subsystems of this DMA, and also the signals from others DMA belonging in the same WDN. Then, the output of the BN is compared with an user-defined threshold to rise or not the

alarm. This work was improved by the application of Expectation Maximization (EM) for the BNs calibration in (Romano et al., 2013a).

In (Jung and Lansey, 2014), a Non-linear Kalman Filter (NKF) is used in combination with pressure and flow measurements inside the network to estimate the nodal demands, which are compared with the values estimated using the NKF. The difference is analyzed using the CUSUM technique and a threshold for leak detection purposes.

In (Ye and Fenner, 2014b), polynomial models are proposed to predict the total weekly water consumption for each measurement using the last week measurements. To make the parameter estimation for these models, an EM algorithm and weighed least squares are used. The residual obtained using the actual measurement is compared with a threshold calculated using the standard deviation when there are no leaks. Once the leak is detected, the difference between the prediction and the real measurement is computed to estimate the leak size.

In (Anjana et al., 2015), a Particle Filter (PF) is used (with a hydraulic model) to reduce the noise in the measurements and then, the CUSUM is applied to detect abnormal consumption (leaks among other events).

In (Hutton and Kapelan, 2015), a polynomial model is calibrated with past measurements of the network water consumption and is used to predict the actual water consumption. Also, the mismatches from the past data (residuals) without leaks are used to create two probabilistic models (one Gaussian, the other heavy tailed, heteroscedastic quantification) for checking (by means of thresholds) if the actual residuals fall into these two models without leaks.

In (Romano et al., 2017), the difference between the current pressure measurements inside the network and the historical ones is compared by using three different statistical methods based on Statistical Process Control (SPC). The outputs of the three tests are then unified in one indicator used to rank the sensors to the most affected until the least. The area surrounding the most affected sensor is the leak area candidate.

2.2.7. Analysis Using Machine Learning

Different approaches to analyze the state of the WDN using the existent machine learning techniques have been studied studied, where usually the analysis tool is trained (not necessary for all techniques) with data (real or artificial). Then, the state of the WDN is compared by the machine learning technique with the trained data (or extracted characteristics) and analyzed.

Several approaches using machine learning have been developed. In (De Silva et al., 2011; Mashford et al., 2012), the pressure or flow measurements are used to classify the new measurements to perform leak localization, size estimation and if leaks exist or not using a classifier based on the Support Vector Machine (SVM) technique with Radial Basis Function (RBF) kernels.

In (Bicik et al., 2011), the evidence theory is used to localize the leak. To do this, the information of three independent sources is used: the Pipe Burst Model Prediction (PBMP) which provides an estimation of the frequency of burst for every pipe, the Consumers Contact Model (CCM) which uses the information provided by the consumers report and the information provided by hydraulic model simulations. After that, the Dempster's rule is used to fuse the output of these three sources and to provide a probability of leak at each pipe where the most probable is the node candidate.

In (Poulakis et al., 2003), the Bayesian reasoning (BR) is used to detect and localize the leak in a benchmark network. Different characteristics are studied (e.g., the effectiveness with pressure or flow measurements, the effects of the different uncertainties and the effects produced by the sensor placement). To do this, IA method is applied, but the estimation of the leak parameters is done by maximizing the probabilities using BR.

In (Xu et al., 2007), a belief rule-based expert system made by experts and trained with past data to improve their performance using Evidential Reasoning (ER) is used to detect leaks in pipe mains using the flow and pressure measurements in the outputs of the pipe main and some pressure sensors inside the pipe main. In (Zhou

et al., 2011), BR is used to update a rule-based expert system parameters instead.

In (Nasir et al., 2010), a Cyber Physical System (CPS) framework is applied to WDNs where the use of Hidden Markov Models (HMM) built with historical data through Bayesian inference allows to evaluate the state of the network (to decide if a leak exists, and if contaminants and water demand are in nominal state).

In (Mounce et al., 2011), Support Vector Regressions (SVRs) are created using pressure and historical flow measurements at the inlet. Then, the prediction of the SVR and the actual measurements are compared and if the occurrence rule is fulfilled, an abnormal behavior (may be a leak) is detected.

In (Goulet et al., 2013), a leak detection and localization method is proposed using model falsification. Different scenarios are created using artificial data (one for each potential leak location taking into account the model uncertainties and noise), and then it is checked if these scenarios are compatible with the current measurements. The ones that are not possible given those measurements are removed from the set of candidate nodes, which at the end (if any remain in the set) provide the detection and the localization result (remaining set of node candidates).

In (Alippi et al., 2013), an ensemble of CDTs is used to detect the time that the leak appears by considering that in healthy conditions the flow measurements at the inlet of the WDN follow an unknown distribution. But, when the leak occurs, then the distribution changes to another one. This approach uses an ensemble of CDTs to be robust against false alarms. In (Boracchi et al., 2013), a similar approach is applied, but instead of using an ensemble a hierarchy of CDTs is used.

In (Boracchi and Roveri, 2014), the periodicity of the water consumption (measurement of the flow at the inlet) is used to apply Self-Similarity (SS) between a set of stored recorded data which represents the current behavior of the WDN in a healthy state and the recent measurements. Over these features, the Intersection-of-Confidence-Interval (ICI) Change Detection Test (CDT) is performed in order to detect a change. If a change is detected, it is assumed that it is due to a leak.

In (Tao et al., 2014), an Artificial Immune System (AIS) network is created using artificial data obtained from a hydraulic simulator and then, the Nearest Neighbor

(NN) method is used to classify the actual data (flow and pressure measurements). Finally, AIS optimization is used to detect and localize the leak.

In (Candelieri et al., 2014), the leaks are considered that occur in the pipes and the number of potential leaks is reduced using flow and pressure measurements (obtained from a hydraulic simulator) and spectral clustering. Then, SVM trained with the clustered data are used to classify the current measurements and perform the leak localization task. A sensor placement method for this specific technique is also presented.

In (Romano et al., 2013b), a model-free technique is presented for the leak localization problem, where four geostatistical techniques including Inverse Distance Weighted (IDW) interpolation technique, Local Polynomial (LP) interpolation, Ordinary Kriging (OK) and Ordinary Cokriging (OC) are presented and tested. These techniques use the pressure measurements inside the network to predict the pressure of each node of the network. Then, the probability that a leak has happened at each node is calculated. Finally, the probability of a leak in each pipe is calculated by computing the average value of the nodes connected by the leak, and the pipe with the largest probability is the node candidate to have the leak.

In (Laucelli et al., 2016), all the measurements (flow and pressure) are used to develop a data model based on the multi-case Evolutionary Polynomial Regression (EPR) to predict the measurements and match them to the current measurements, and then the residuals are compared with thresholds to detect leaks.

In (Mounce et al., 2014), a database of normalized inlet flow patterns coming from different networks is used to retrieve the most similar pattern using k-Nearest Neighbors (k-NN), to compare the degree of similarity against a threshold for abnormal consumption detection. This approach was extended in (Mounce et al., 2015) to exploit the measurements coming from Automatic Meter Readers (AMRs) with the use of Big Data and Cloud computing technologies.

In (Kim et al., 2015), a leak detection and localization technique is proposed. The detection uses the inlet flow measurements at a high sampling rate where a KF is applied to remove the noise. Then, the mean of the signals is removed using a

time window and a floor function, for finally, comparing the resulting value against a threshold and check if it is overpassed a determinate number of times in a short period of time. After that, a leak starting time is estimated using the same indicator but using instead the pressure measurements placed inside the network and searching for the moment where the curvature in the function is maximum. The different leak starting time along with the position of the sensors are used to triangulate the leak location.

In (Rajeswaran et al., 2017), a leak localization technique is proposed using a graph portioning algorithm. The WDN, which is considered that all the water inflows and outflows are measured (except the leak), is modeled as a graph. Then, when the measured sum of water inflows and outflows mismatch, the leak is detected and the graph algorithm finds the best partition of the network to have two zones as equal size as possible, but at the same time with as fewer connections possible. Then, flow meters are placed in pipes that connects the two zones and the zone with the leak can be identified with those measurements. Once this is done, half of the network is discarded as a potential leak location zone, and the process is repeated again until the leak is localized in a pipe.

Principal Component Analysis (PCA) Techniques

In the field of leak detection, the PCA technique is applied (see (Palau et al., 2012)) using the historical flow measurements at the inlet of the DMA to compute the PCA model with the aim of reducing the historical data to the most relevant components. Then, the PCA model with the new flow measurements is analyzed by the application of a statistic test (T^2 Hotelling and Distance to MODel (DMOD) are presented) and analyze if the new data differs enough to determine if a leak exists or not.

In (Nowicki and Grochowski, 2011), a Kernel Principal Component Analysis (KPCA) is used to detect the non-linear patterns of the pressure and flow measurements in the WDN, and then is evaluated if the new measurements differ enough to consider that a leak is present in the WDN.

Another PCA approach (Gertler et al., 2010) uses the WDN measurements (flow

and pressure at the inlet and pressure inside) to build a PCA model, which is used with the current measurement to check if any load exceed a predefined threshold, which will mean that a leak is detected.

In (Sánchez-Fernández et al., 2015), four different PCA approaches are proposed to detect a leak (and contaminants in the water) in WDNs using pressure and flow measurements. The network is discretizated in several zones. One of the PCA technique uses the classical PCA scheme, which consider all the network (the sectorization is not used in this case) to perform the analysis. The others use the zones created and different Distributed PCA (DPCA) techniques to analyze them. The output of the PCA model with the new measurements is analyzed by the T^2 and the Q statistic, which are also compared, to perform the task of leak detection.

In (Quiñero Grueiro et al., 2016), the repetitive flow pattern consumption is exploited to create a cyclic PCA (different PCA models are build depending on the different hourly pattern consumption of water) technique using the pressure sensors inside the network, where the statistics T^2 and the Squared Prediction Error (SPE) are used and compared, to detect leaks.

Fuzzy-Based Techniques

In (Wachla et al., 2015; Moczulski et al., 2016), neuro-fuzzy classifiers are used to detect and localize the leaks in a discretized network (i.e., the networks are reduced to discrete zones) using flow measurements. Similar approach is considered in (Sanz et al., 2012) using pressure measurements instead.

In (Li and Li, 2010), pressure measurements are used to perform pipe clustering depending on the similarity of the pressure at the area (to know the pressure a hydraulic simulator is used). Then, using the mismatch with those expected values and the current pressure measurements, a fuzzy recognition system is applied to detect and localize the leak into the area that mismatch the expected behavior.

In (Islam et al., 2011), the measurements (flow and pressure at the inlet and pressure inside the WDN) are fuzzyficated using historical records. Then, the new measurements are compared with these values to detect if the limits are crossed over time,

what allows to detect the leak. The sensor that indicates the leak (or the sensor most affected if the threshold of more than one measurement is crossed) is used to localize the leak in the area where the sensor is placed.

Artificial Neural Networks (ANN) Techniques

In (Daoudi et al., 2005), the leak is detected using the information extracted from acoustic measurements and wavelets, then PCA is applied to reduce dimensionality, and finally, measurements are classified using a Multi-Layer Perception (MLP) ANN (before that, the ANN is trained with representative data measurements) as leak or not leak.

In (Mounce et al., 2003), an ANN with feedforward MLP is used to build a model of Gaussian Mixture Models (GMMs), which results in a network that provides the conditional probability density of the output data, which is called Mixture Density Network (MDN). This is done for every sensor data. At the output of the MDN, a rule-based expert system is applied to classify the output as being in normal or abnormal state. This system is presented in a WDN with some DMAs with flow sensors in the inlets. This approach is able to detect and localize the affected DMA with the flow measurements, and the node affected in a DMA with the pressure sensors.

In (Caputo and Pelagagge, 2003), a MLP back propagation ANN (trained with simulated data, considering that there are flow sensors at every pipe and pressure sensors at every node) is used in a hierarchical way. First, the leak is detected, and in the second step the leak is localized and their size estimated.

In (Yang et al., 2008), a leak detection method is proposed where the SS and the Approximate Entropy (ApEn) algorithm are used to extract features from acoustic measurements coming from different sources (leak, cars, machines, etc.). These extracted features are used to train a two layer ANN Elman Network (EN) (back propagation training) to classify the features from new measurements. This is only applicable to pipe segments. Later, in (Yang et al., 2010) a leak detection in pipe mains is proposed using the acoustic sound measurements, which are processed to

extract features using ApEn algorithm. Historical records of leak and no leak events are used to train a two layer ANN EN which classifies the current measurements into leak or no leak classes.

The problem of leak detection is treated using an ANN of type Self-Organized Map (SOM) using data with and without leaks (the data is not labeled) and a function that describes the leak. The output provided the probability of the leak in the range from 0 to 1, where a threshold is used to decide if there is enough evidence to raise an alarm. This approach is presented in (Aksela et al., 2009).

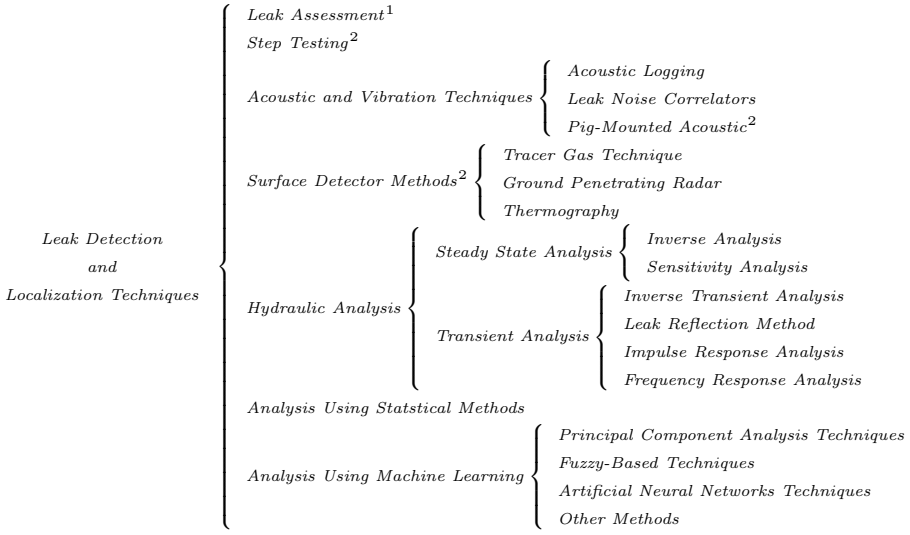
In (Mounce et al., 2010), a MDN ANN is trained using flow measurements at the inlet through back propagation. The mismatch of the ANN prediction with the current measurements (in a time horizon) is fed to a fuzzy inference system, which classifies the current flow consumption as normal or abnormal (possible leak).

In (Zhang and Wang, 2011), the Bayesian theory is used to model the leaks (which for each potential leak location, considered at pipes, the probability of leak at the pipe and the leak size are estimated) as a Probability Density Function (PDF) with Gaussian distribution, where the Gaussian parameters are estimated using Fisher's Law in order to minimize the difference with respect of the considered measurements, flow and pressure measurements inside the network. If any modeled leak present a large probability over the others, the detection alarm is raised, and it is assumed that the leak is at the pipe that activates the alarm. Then, a leak prediction using a back propagation ANN is used to estimate the leak size computing the difference between expected water consumption predicted by the ANN and the measured total water inflow.

In (Rojek and Studziński, 2014), two different ANN for leak localization using flow measurements are compared. On the one hand, MLP ANN with back propagation network (the work concludes that is the best of the two) and, on the other hand, an ANN Kohonen network. The two ANNs are trained and fed with the residuals generated with the measurements and the expected values obtained through a hydraulic simulator.

2.2.8. Summary of Leak Detection and Localization Techniques

The leak detection and localization techniques presented in this section can be grouped and summarized in the following scheme.



where the superscript [1] means that the technique is only applicable in the leak detection task while the superscript [2] that the technique is only applicable for leak localization purposes.

2.2.9. Sensor Placement Techniques

Due to the budget restrictions, the number of sensors that can be deployed into the networks is limited (usually, the pressure sensors are placed at nodes and the flow rate sensors at pipes). So, the sensor placement techniques aim to obtain the best leak localization (and in some cases also detection) performance given the method by placing the sensors into the WDN in an intelligent and efficient way.

The main approach to sensor placement is by formulating an optimization problem that involves minimizing (or maximizing) an objective function that includes some criteria related to the method which is designed for. The objective function is

usually a non-linear integer problem and the application of an exhaustive solver is not realistic in practice because of the computational cost.

One way to overcome this difficulty is by using Genetic Algorithms (GAs), see (Casillas et al., 2013) for the case of the angle method, and a more elaborate procedure, using Particle Swarm Optimization (PSO), in (Casillas et al., 2015b). An example of the different approaches in the optimal sensor placement can be seen in (Vítkovský et al., 2003) where different approaches are considered by including different objectives, the amount of data used and the sampling rate. In (Blesa et al., 2014), the uncertainties of the model are handled using a clustering technique (Evidential c-means) that is applied over the FSM to reduce the potential number of places to install the sensors by clustering sensors with similar behavior. Then, for each cluster a representative is chosen and the optimization problem is solved using the Branch and Bound algorithm. In (Steffelbauer and Fuchs-Hanusch, 2016), the leak sensitivity matrix is improved by taking into account the demand uncertainty through Monte Carlo simulations, to find the best sensor locations using GA.

In (Pérez et al., 2014), the impact of adding more sensors for the correlation FSM method proposed in (Quevedo et al., 2012) is discussed.

In (Sarrate et al., 2012), a sensor placement for the structural model-based diagnosis is proposed, where a leak isolation index is optimized by means of the Branch and Bound algorithm.

In (Wysogląd and Wyczółkowski, 2007), a case of leak detection and classification using ANN is presented along with a sensor placement for the proposed technique using GA and the classification accuracy as objective function.

In (Narayanan et al., 2014a), a method for the case that the hydraulic model is not known is presented, but instead an estimated graph model is used considering only the topological information and central metrics. From this graph model, the pressure in the nodes, the flow rate in the pipes and the burst probability in the pipes are calculated and ranked. Then, some sensors are placed on the nodes appearing the highest positions of the ranking.

In (Giorgio Bort et al., 2014), two pressure sensor placement methods are presented

for the sensitivity analysis in steady sate by using the percentage of pressure change in every node for each considered leak. Then, the features of the hourly and daily mean and variance of the percentage of pressure change are obtained. The first method is to apply PCA on those features and use the coefficients of the first component to rank the nodes where the sensors must be placed (i.e., if there are five sensors to place, the first five nodes in the ranking according to the weights of the first principal component are the ones chosen to place the sensors). The second approach uses the positive coefficients of the PCA to perform an optimization problem where the fitness function is the minimum correlation between them.

In (Cugueró-Escofet et al., 2015a), an index to evaluate the quality of sensor placement based on the confusion matrix is presented to take into account the clustering of similar leaks by their geographical location, and then using the GA to solve the posterior optimization problem.

In (Meseguer et al., 2015), a real sensor placement is carried out, where a sensor placement for the correlation method (the technique is presented in (Quevedo et al., 2012)) solved using GA is done in a real network is presented and evaluated. Another real sensor placement is presented in (Farley et al., 2013), for the method presented in (Farley et al., 2010), where a sensitive matrix with the information of all sensors (obtained using a hydraulic simulator) is exploited by the GA to solve the optimization problem.

In (Sanz and Pérez, 2015), a WDN calibration work based on the Single Value Decomposition (SVD) is presented for the estimation of the parameters and the demand components. But the SVD is further exploited to detect the locations where the sensors have large sensitivity to one demand and low for the others, which are good places to install the sensors for the leak detection and localization method presented in the same work and in (Sanz et al., 2015).

In (Sarrate et al., 2016), metrics related to the leak localization Angle method are used to cluster the nodes in a few zones. Then, a small set of representatives nodes for each cluster are selected using a semi-exhaustive search. While the combination of sensors is evaluated, if at any moment the indicator of this current placement has

no options to be better than the best tested so far, the procedure is stopped and the next combination is evaluated.

In (Perelman et al., 2016), a sensor placement that maximizes the leak isolation of the FSM is presented where an augmented greedy minimum set cover is proposed to find the best suitable places to install the sensors.

2.3. Validation and Evaluation Indicators

In this section, different indicators will be introduced to evaluate the performance of the methods presented, focused on the problem of leak detection and localization.

2.3.1. Leak Detection Indicators

To assess whether a fault detection method is working properly the Type I and Type II errors are used, also known as False Positive Rate (FPR) and False Negative Rate (FNR). Here the sequences to test the leak detection performance are composed by several days of measurements without leak and some more days with leak, so, in every sequence there is a leak after some time without leak. The FPR tells us the percentage of erroneous detections, i.e., no leak is present in the network when the detection is raised, of the total number of sequences analyzed. The FNR tells us the percentage of omitted detections, i.e., when a sequence with a leak has occurred ends without raising a detection, of the total number of sequences analyzed.

Apart of these two indicators, the Detection Delay (DD) is one of the more important, which tell us how long (in hours) the detection method requires, in average, to raise a correct detection.

Since the leak detection method presented in Chapter 4 has the ability to estimate the time instant in the time series where the leak has started, the Difference Time Detection (DTD) is used. This indicator measures the time in hours between the

real moment when the leak has start and the estimated one by the leak detection technique tested.

Finally, the leak size is estimated as a part of the leak detection procedure, from which we define Δl as the difference between the estimated leak size by the leak detection method tested and the real leak size in [l/s].

2.3.2. Leak Localization Indicators

One of the most used indicators in Fault Detection and Isolation (FDI) that summerizes the results of the application of a given method to a complete set of leak scenarios is the Confusion matrix ($n_c \times n_c$) depicted in Table 2.1, where n_c is the number of classes (assuming that each class corresponds to a different kind of fault (leak l)). The rows represents the leak scenario and the column corresponds to the leak is localized ($\hat{l}$) by the considered approach.

Table 2.1: Confusion matrix Γ.

	$\hat{l}_1$	$\cdots$	$\hat{l}_i$	$\cdots$	$\hat{l}_{n_c}$
l_1	$\Gamma_{1,1}$	$\cdots$	$\Gamma_{1,i}$	$\cdots$	Γ_{1,n_c}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
l_i	$\Gamma_{i,1}$	$\cdots$	$\Gamma_{i,i}$	$\cdots$	Γ_{i,n_c}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
l_{n_c}	$\Gamma_{n_c,1}$	$\cdots$	$\Gamma_{n_c,i}$	$\cdots$	Γ_{n_c,n_c}

In case of a perfect classification, the confusion matrix should be diagonal, with $\Gamma_{i,i} = m_V$, for all $i = 1, \ldots, n_c$ being m_V the size of the validation data set (number of examples in each class) and n_c the number of classes. In our case, according to the previous section, the number of classes n_c is the number of places of the WDN where a leak is considered, i.e., the number of network nodes n_n. But this number could be changed for example removing the data from nodes that are very close to others and reducing the number of potential leaks (classes).

In practice, non-zero coefficients will appear outside the main diagonal of matrix $\mathbf{\Gamma}$. For a leak in node i, the coefficient $\Gamma_{i,i}$ indicates the number of times that the leak l_i is correctly identified as $\hat{l}_i$, while $\sum_{j=1}^{n_c} \Gamma_{i,j} - \Gamma_{i,i}$ indicates the number of times

that it is wrongly classified. The overall accuracy Ac of the classifier is defined as

$$\mathrm{Ac} = \frac{\sum_{i=1}^{n_c} \Gamma_{i,i}}{\sum_{i=1}^{n_c} \sum_{j=1}^{n_c} \Gamma_{i,j}} \tag{2.16}$$

The term "node relaxation", used later, refers to the minimum 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.

Since the accuracy value (2.16) only provides a reference of the classification goodness and not how good it is the leak localization, the Average Topological Distance (ATD) is the indicator used to assess the overall performance. The ATD is the average value of the the minimum distance in nodes between the node with the leak and the node candidate proposed by the leak localization method. The ATD is computed as follows

$$\mathrm{ATD} = \frac{\sum_{i=1}^{n_c} \sum_{j=1}^{n_c} \Gamma_{i,j} \mathrm{D}_{i,j}}{\sum_{i=1}^{n_c} \sum_{j=1}^{n_c} \Gamma_{i,j}} \tag{2.17}$$

where $\mathbf{D}$ is a symmetric square matrix with size n_c such that each element $\mathrm{D}_{i,j}$ contains the minimum topological distance in nodes between the nodes referred by indices i and j.

2.4. Machine Learning

In this section, a brief introduction to machine learning and their basic terminology is presented, along with the techniques of this field applied in this PhD thesis.

There are many descriptions of what machine learning is, for example, "Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that machines should be able to learn and adapt through experience"[1].

[1] https://www.sas.com/en_us/insights/analytics/machine-learning.html

2.4.1. Basic Terminology

Some of the basics concepts and terms used in this PhD thesis and related to machine learning are summarized and introduced here (Mohri et al., 2012):

- **Examples**: are the instances of data used in the learning and evaluation stages.
- **Attributes**: are the set of variables or features used to learn from the data.
- **Labels**: are the different classes where the new instances (or examples) may be assigned.
- **Training data set**: is the set of examples used to train the algorithm.
- **Validation data set**: is the set of examples used to select the appropriated free parameters in the algorithm.
- **Test (or testing) data set**: is the set of examples used to evaluate the performance of the algorithm.

2.4.2. Kinds of Techniques and Learning Methods

The machine learning techniques can be divided taking into account the objective in (Mohri et al., 2012):

- **Classification**: assign instances to labels (categories).
- **Regression**: predict a real value for each instance.
- **Ranking**: order instances using some criteria.
- **Clustering**: grouping instances into labels.

- **Dimensionality reduction**: reduce the initial representation of instances into one with lower dimensionality while preserving some properties at the same time.

When considering the method used to learn, the following methods exist:

- **Supervised learning**: the learner receives labeled examples as training data and make prediction for new unlabeled instances.
- **Unsupervised learning**: the learner receives unlabeled examples and make predictions for new unlabeled instances.
- **Semi-supervised learning**: the learner receives labeled and unlabeled examples and predict for new unlabeled instances.
- **Transductive inference**: the learner receives labeled and unlabeled examples and predict only for the unlabeled examples introduced.
- **On-line learning**: the learner receive an unlabeled example, then makes a prediction and receives the true label of this example, calculate the error to adapt the algorithm and the process is repeated until the training samples are finished or the error criterion is achieved.
- **Reinforcement learning**: the learner evolves in time gaining experience by receiving a reward according to their actions giving instance (i.e., output of the technique).
- **Active learning**: the learner continuously adds new examples in their training data set using the new instances labeled by an oracle.

2.4.3. Semi-Supervised Learning

The semi-supervised learning deals with a training data set of labeled data, in which the new unlabeled instances are revived to assess if this new instance has enough evidence to belong to the classes in which the learner has been trained, or belongs to a new class.

Intersection-of-Confidence-Interval Change Detection Test

The ICI CDT technique uses data from an unique class, also known as in control, where the characteristics of the observed signals are considered in normal state, which in the case of the ICI CDT means that are in an stationary state. Then, the new measurements are evaluated to decide if they belong to the same class or into a new one, which this new class is known as out of control.

The ICI CDT uses the intersection of intervals to assess if a change has occurred. These intervals are computed using the mean and the standard deviation, which first are computed using the training data set as

$$\mu_T = \sum_{t=1}^{m_T} \frac{x(t)}{m_T} \tag{2.18}$$

where μ_T is the mean for the training data set, m_T is the training data set size and $x(t)$ is the feature to monitor. It is considered that the time instant where the technique starts to analyze the new data is $t = m_T + 1$. One the other hand, the standard deviation for the training data set σ_T is obtained computing

$$\sigma_T = \sqrt{\sum_{t=1}^{m_T} \frac{(x(t) - \mu_T)^2}{m_T - 1}} \tag{2.19}$$

Once the initials μ_T and σ_T are computed, the recursivity updated values of μ and σ are calculated as

$$\mu(t) = \frac{(t - \iota)\mu(t-1) + \sum_{i=1}^{\iota} x(t - \iota + i)}{t} \tag{2.20}$$

where ι is the size of the data added (i.e., the window size to calculate a new interval) to calculate the updated $\mu(t)$ at time t for any $t > m_T$. Note that in the first interval after the training ($t = m_T + 1$), $\mu(t-1) = \mu_T$. And

$$\sigma(t) = \frac{\sigma_T}{\sqrt{t}} \tag{2.21}$$

again, for any $t > m_T$.

Then, the interval $\mathcal{I}(t)$ is computed using these two values, where the upper bound is

$$\mathcal{I}^{(+)}(t) = \mu(t) + \Upsilon\sigma(t) \tag{2.22}$$

where the Υ is a user defined value (usually between one and five depending on the desired trade-off between the FPR and the FNR), and the lower bound

$$\mathcal{I}^{(-)}(t) = \mu(t) - \Upsilon\sigma(t) \tag{2.23}$$

So, the ICI CDT detects a change when the the new computed interval does not intersect with at least one of the previously computed intervals. Once the detection is raised, the time instant where the change has started can be estimated. To do that, the same technique is applied over the same feature $x(t)$ but using a smaller Υ value. This allows to detect an incipient change in the time series, and by knowing that there is a change in them, assume that this new detected changes is the estimated change starting time.

One example of the ICI CDT technique can be seen in Figure 2.2. Note that the change detection time is at the end of the interval where the interval do not intersect with the others while for the estimated change starting time is at the beginning.

This technique is used to catch the behavior of nominal conditions of the total flow consumption. Then, it detects when a new behavior appears and analyzes if it is due to the presence of a leak inside the network or not.

2.4.4. Supervised Learning

This kind of machine learning techniques uses labeled data (training data set) to find a function of mapping (classification) or to search relations among variables (regression).

Formally speaking, the supervised learning uses a training data set $\langle x_i, y_i \rangle$ where x_i is an array of input values and y_i is the output to find a function $fun : X \to Y$.

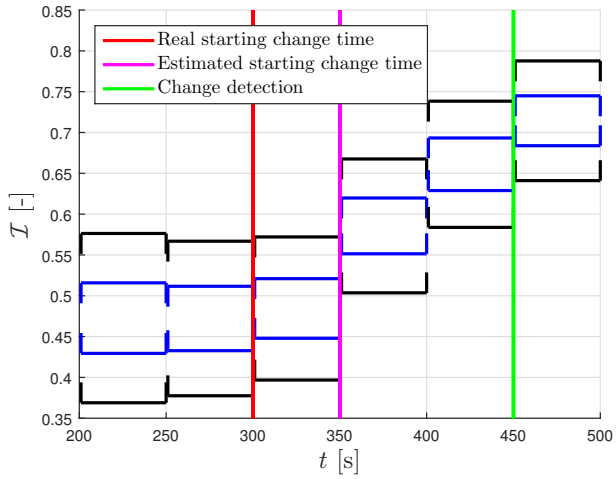


Figure 2.2: Example of the procedures of change detection (black intervals) and estimated change starting time (blue intervals).

In this PhD thesis, three techniques of supervised learning are used. On the one hand, two classification techniques which are the k-Nearest Neighbors and the multi-class Bayesian classifier are used. On the other hand, the Kriging interpolation is used as a regression technique.

The k-Nearest Neighbors Classifier

One of the well accepted and established method for classification is the k-Nearest Neighbors (k-NN) algorithm (Alpaydin, 2010), which is available in most of the numerical packages (e.g., Matlab, R, etc.). Its basic version works as follows. When a new data instance has to be classified, the distances[2] to all the instances in the training data set are computed. Then, the k nearest neighbors are selected and a voting procedure is applied, where each neighbor votes for its own class and the class with more votes is chosen as the associated class for the new data instance. The process is illustrated in Figure 2.3, where a value $k = 3$ is used and the new data instance is associated to the class cl_3 since two of the three minimal distances are associated to training instances in that class.

The use of a value for k bigger than one improves the robustness against outliers

[2]Typically, the Euclidean distance is used, but many other metrics can be used.

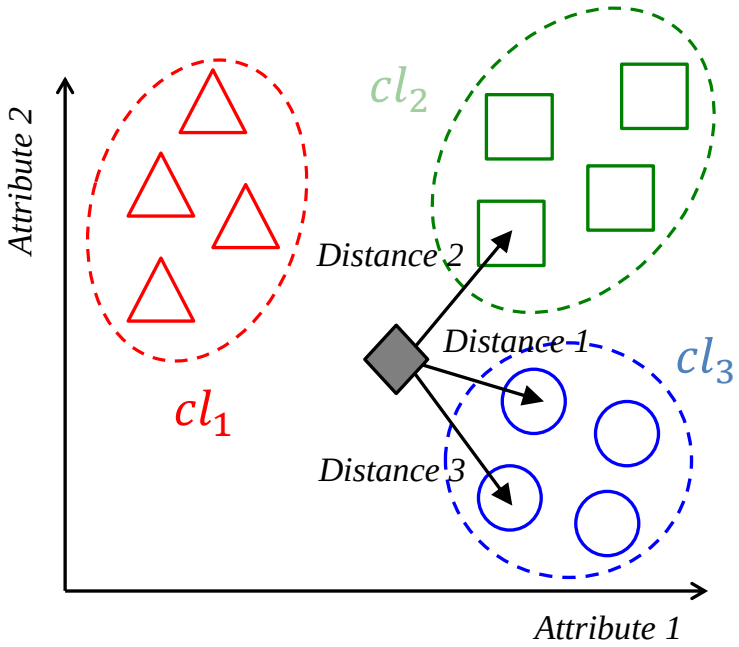


Figure 2.3: The k-NN algorithm.

(with $k = 1$ the class of the nearest neighbor is selected, which seems a good choice, but the obtained classifier is really sensitive to outliers). On the other hand, the value for k must be smaller than the minimum number of instances associated to a single class inside the training data set.

Temporal reasoning

The classification problem deals with a specific instance to be classified each time, and instances from different time instants does not have to belong to the same class. But in the specific problem of leak localization, the leak does not change the place in the networks where it has been generated. This means that the stream of instances to be classified belong to the same class. This particular fact can be exploited with a temporal reasoning to enhance the diagnosis performance since the performance of the classification with a direct application can provide poor classification results if the classes have uncertainties that lead to overlap their space with other classes. To smooth the effect of this overlapping, the classification in a time horizon with length N is proposed.

A simple temporal reasoning can be based on taking into account the estimations

provided by the classifier inside the time horizon and applying a voting scheme, concluding that the candidate is the class that more times has been selected by the classifier.

A second and more sophisticated option could be to use the information contained in the confusion matrix. Hence, at each time instant t, when the classifier is providing a class candidate as an explanation for the instances in the current time instant t, the whole column j of the confusion matrix is stored. This column provides an estimation of the probabilities $P(cl_i|\hat{cl}_j)$, i.e. the probabilities that the true class i is classified as class j, according to the available information available for current time instant t. Then the sum of column vectors stored along the time horizon N as $t-N+1,\ldots,t$ is computed. In the obtained vector, the position of the coefficient with highest value indicates the most probable class according to the information provided by the data in the whole time horizon $t-N+1,\ldots,t$.

The Multi-Class Bayesian Classifier

The Bayesian reasoning gives the probabilities that the instance analyzed belongs to every possible class using the Bayes rule at every time instant t

$$P(cl_i \mid \mathbf{x}(t)) = \frac{P(\mathbf{x}(t) \mid cl_i)P(cl_i)}{P(\mathbf{x}(t))} \text{ for } i = 1,\ldots,n_c \tag{2.24}$$

where n_c is the number of classes, $P(cl_i \mid \mathbf{x}(t))$ is the posterior probability that the new instance $\mathbf{x}(t)$ belongs to the class cl_i, $P(\mathbf{x}(t) \mid cl_i)$ is the likelihood of the instance $\mathbf{x}(t)$ assuming that the class is cl_i, $P(cl_i)$ is the prior probability for the class cl_i, and $P(\mathbf{x}(t))$ is a normalizing factor given by the Total Probability Law

$$P(\mathbf{x}(t)) = \sum_{i=1}^{n_c} P(\mathbf{x}(t) \mid cl_i)P(cl_i) \tag{2.25}$$

Without previous information, the prior probabilities used are considered to be equal for all classes. This assumption changes when taking into account a time horizon N diagnosis. Then, the prior probability in the time instant t is the posterior probability of the previous instant $(t-1)$ except the first diagnosis, which can be

formulated as

$$P(cl_i \mid \mathbf{r}(t-N+j)) = \frac{P(\mathbf{x}(t-N+j) \mid cl_i)P(cl_i \mid \mathbf{x}(t-N+j-1))}{P(\mathbf{x}(t-N+j))} \quad \text{for } i = 1,\ldots,n_c \text{ and } j = 1,\ldots,N \tag{2.26}$$

where a starting point with all classes with the same prior probability $P(cl_i \mid \mathbf{x}(t-N)) = \frac{1}{n_c}$ for $i = 1,\ldots,n_c$.

One drawback of using this time horizon analysis is related to the fact that one class can take a probability value close to one, and then the rest of probabilities remain close to zero. To prevent this, the posterior probabilities are forced to a maximum value, such that if one result overpasses that value, then the difference is equally distributed to the others classes.

These techniques are used to better infer the diagnosis using the residuals generated in the model-based leak localization methodologies. Also, the Bayesian temporal reasoning is used to improve the diagnosis in the proposed data-driven leak localization approach.

Kriging Interpolation

The Kriging interpolation is a multi-variate regression that uses a few measured points with their position in the space to find a spatial model that exploits the relations among different points using their locations. There are different Kriging models in the literature, here the on implemented in the DACE toolbox (Lophaven et al., 2002) is used. So, a new point in the same space can be estimated given a new location $\mathbf{x}$ and the estimated spatial model as

$$\hat{y}(\mathbf{x}) = \mathfrak{H} + \varepsilon\left(\boldsymbol{\chi}, \boldsymbol{\theta}, \mathbf{s}\right) \tag{2.27}$$

where $\hat{y}$ is the value of the interpolation, $\mathfrak{H}$ is a constant that represents the constant part of the interpolation and function $\varepsilon(\boldsymbol{\chi}, \boldsymbol{\theta}, \mathbf{x})$ is the spatially correlated part of the variation. Both terms Constant $\mathfrak{H}$ and function $\varepsilon(\cdot)$ are obtained in the fitting

process as well as function parameters $\boldsymbol{\chi}$ and $\boldsymbol{\theta}$. On the other hand, $\mathbf{x}$ is the location of the new point where the interpolation is made. The fitting process consists in a least square error minimization problem considering available locations with data $\mathbf{y}_T$ and their locations $\mathbf{x}_T$. Function $\varepsilon(\cdot)$ has two parts to be estimated in the fitting process

$$\varepsilon\left(\boldsymbol{\chi}, \boldsymbol{\theta}, \mathbf{s}\right) = \tau\left(\varsigma fun_p(\boldsymbol{\chi}, \mathbf{x}) + \varpi fun_c(\boldsymbol{\theta}, \mathfrak{l})\right) \tag{2.28}$$

On the one hand, the polynomial function $fun_p(\cdot)$ whose arguments are $\boldsymbol{\chi} \in \Re^u$ where u is the polynomial order plus one and $\mathbf{d}_i(\mathbf{q})$. And on the other hand, the correlation function $fun_c(\cdot)$ whose arguments are $\boldsymbol{\theta} \in \Re^m$ where m is the dimension of the variogram, i.e., the number of the dimensional space of the interpolation map and also $\mathbf{x}$. Finally, τ is a scaling factor obtained in the fitting process as $\mathfrak{H}$, ς and ϖ. The last two parameters are estimated constants to balance between the polynomial and the correlation functions.

The other supervised learning technique is used for the leak localization task but avoiding the use of hydraulic models to create a purely data-driven approach.

2.4.5. Feature Selection

Feature Selection (FS) is one of the dimensionally reduction problems, where by using a subset of features of the total set, the feature selection algorithm aims to preserve the maximum amount of information contained in the original set of features.

There are four main categories of FS techniques recognized in the literature (Saeys et al., 2007; Bolón-Canedo et al., 2013): filter based methods, wrapper methods, embedded methods and, finally, hybrid methods, i.e., combination of filters with wrappers. The methods of the first type, filter based methods (Vergara and Estévez, 2014), directly work with the data, without interacting in any way with the model to be built. Hence, individual features or feature sets are evaluated according to some metrics that are assumed to be fast to be computed. Some of the most com-

mon indicators are the relevance, i.e., the information contained in a given feature (according to the final application) (Guyon and Elisseeff, 2003; Chandrashekar and Sahin, 2014), and the redundancy, i.e., how much of the information in a given feature is repeated in others (Salmerón et al., 2016; Liu et al., 2016). Many existing filter methods combine these two indicators (Yu and Liu, 2004; Peng et al., 2005). The main advantage of this type of methods is their low computational cost, while the main drawback is that the selection does not take into account the posterior use of the data by the model. The second type of methods, wrapper methods (Chandrashekar and Sahin, 2014), need to build and use the model to score selected feature subsets that are generated within the framework of an heuristic search. Some methods in this category are based on the use of GAs (Oreski and Oreski, 2014) and on PSO methods (Xue et al., 2013), among others. Due to the search and to the fact that a new model has to be trained (build) for each subset, these methods are computationally demanding, but they usually provide the best results for the particular type of model used. Embedded methods are the third type of methods, and they combine the use of the model that ranks the features in a priority order to be selected. In this group, there are techniques such as Backward Feature Selection (BFS) (Guyon and Elisseeff, 2003), Random Forest (RF) (Díaz-Uriarte and De Andres, 2006) and, in general, Evolutionary Algorithms (EA) (Xue et al., 2016). Finally, the most recent approaches are the hybrid methods, which typically combine a filter that reduces the initial number of features with a wrapper that provides an additional refinement (Inbarani et al., 2014; Hu et al., 2015; Apolloni et al., 2016). The latter approach is considered in the present work due to the obtained good compromise between optimality and computation time for the classifier-based leak localization techniques while a embedded method is used for the data-driven leak localization approach.

Fast Correlation-Based Filter (FCBF)

The FCBF presented in (Yu and Liu, 2004) is a feature selection algorithm used to select a subset of features that takes into account the relevance of the features and

the redundancy between each pair of features in an efficient way. An example of this technique is depicted in Figure 2.4.

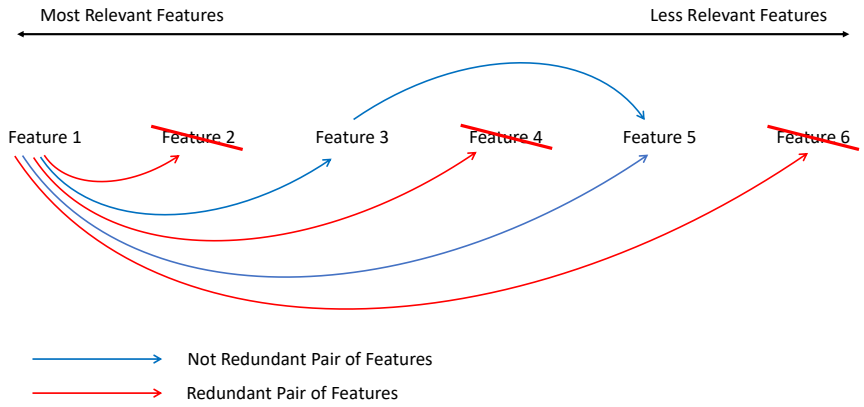


Figure 2.4: Example of the fast correlation-based filter.

First, the relevance of each feature is evaluated according to a particular metric. Once the relevance of each feature could be calculated, the redundancy of each pair of features is evaluated, but to avoid the necessity of assessing the redundancy of each pair of features, the FCBF algorithm is applied to avoid unnecessary computations. Starting from the most relevant, this feature is compared with the rest in a descending order of relevance, with the other features to evaluate their redundancy. If their redundancy is larger than a predefined threshold, then the less relevant of the two features is discarded. Once the most relevant feature is compared with all the features, the next most relevant and not discarded feature is compared with all the remaining features less relevant than them which are not discarded yet. This process is repeated until all the remaining features are evaluated with each other. The surviving set of features has the properties of being the most relevant and present low redundancy between them. Note that the final number of features can be controlled with the predefined threshold.

Sequential Forward Floating Selection (SFFS)

The SFFS is a feature selection technique presented in (Pudil et al., 1994) that adds in an incremental way a new feature that best improves the current subset

of features according to an objective function. This particular algorithm has the ability of move backwards (i.e., remove features if a new, smaller that the actual, subset can be obtained that the one obtained before by that number of features). The SFFS has two main parts at every iteration that are computed in a sequential manner. In the first part, a forward selection of one step is done, meanwhile in the second part, a backward selection of one or more steps is done if the conditions are fulfilled. This process is depicted in Figure 2.5.

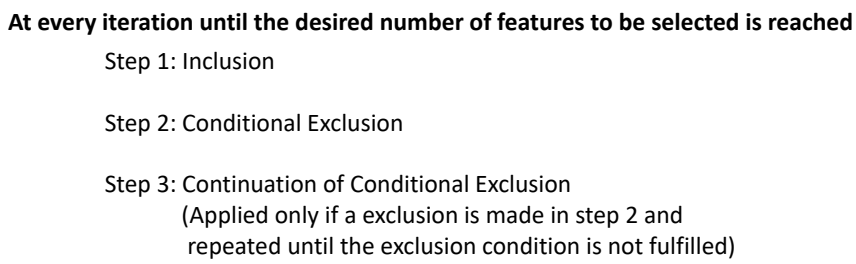


Figure 2.5: Sequential Forward Floating Selection.

At every iteration, first all the potential features to be added to the actual subset are individually tested by adding them to the actual subset and by computing the fitness function. The feature that provides the best performance is then added to the subset of features. Then, in the second part of the algorithm and if at least there are two features in the subset, all the features that are in the subset of selected features are removed individually (i.e., one at a time) and the fitness function is tested. If the best value according to the fitness function obtained is better (note that by better means that the last added sensor never will be removed in the first step of the second part of the iteration) than the best one obtained before for that number of features, the feature is removed, and the process of trying to remove features is continued until there is no better subset of features than the ones obtained before or there is only one feature in the subset. The algorithm ends when the at the end of the iteration there are the number of features that is wanted to be selected.
These different methods are applied to the problem of sensor placement to tackle the combinatorial problem leading to a feasible approach.

2.4.6. Optimization

Optimization algorithms aim to learn from previous optimization iterations to apply an intelligent change on the parameters being optimized to obtain closer values to the optimum given a fitness function until one of the criteria to stop the optimization process is reached.

Genetic Algorithm

A GA provides an optimization engine based on the genetic evolution, where starting from a seed population (first generation) with a fixed population size p_s, each member of the population is evaluated according to a fitness (or objective) function and ranked. The best ones (their number is determined by the elite count parameter e_c) survive to the next generation, and the remaining members of the new generation (until the p_s number) are members derived from the ones that have survived. The process is repeated until one of the stopping criteria is accomplished, for instance, the maximum number of generations max_g is reached, or no best member has been found from one generation to the next (i.e., the difference is less than a tolerance tol).

The GA implementation works as follows. The members of the first generation of population are randomly created by the GA (also can be introduced by the user) with the specified population size p_s and then evaluated to select the best ones. The next generation of population is obtained from the elite members of the previous one, which directly pass from one generation to the next, and a number of filling members created by a pool onto the former best members. These filling members are identified by means of a bit string and are chosen by a tournament selection with the application of Laplace crossover (fusion of two members by swapping parts of their bit strings) and power mutation (when a bit in the string that define the members is changed) with a truncation process to ensure integer members. For further details see (Deep et al., 2009). This technique is graphically explained in Figure 2.6.

It should be noted that the objective function that is wanted to be minimized is

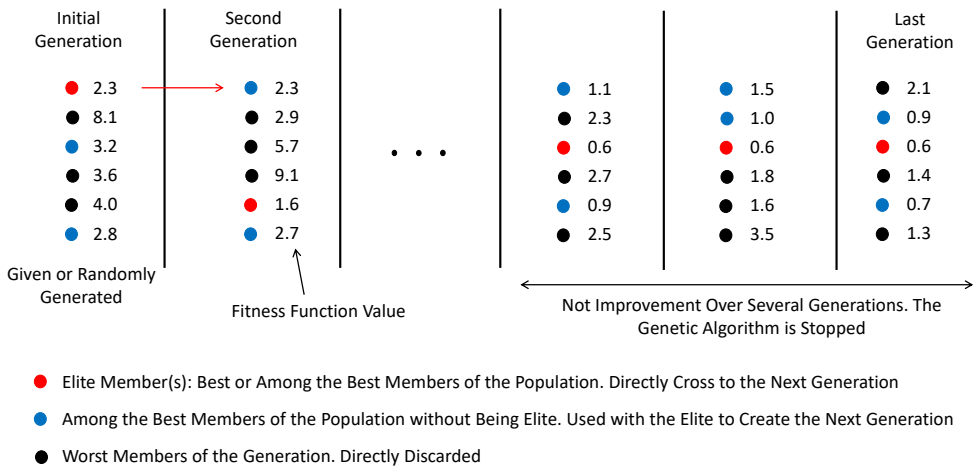


Figure 2.6: Genetic algorithm.

not the one that the implemented version in Matlab and used here does. This implementation first prioritize that the combination tested fulfill the constraints. If any member of the population achieves this goal, then the objective function introduced by the user is the one that is applied.

Like the feature selection techniques, the proposed optimization approach is used to solve an integer optimization problem in the sensor placement task.

3. Case Studies

In this chapter, the case studies used along the PhD thesis are presented. In particular, the WDN and DMA networks are detailed and the real cases associated to them are also described.

First, the Barcelona DMAs with only flow measurements at the inlets are presented, which are the ones used to test the proposed leak detection technique proposed in Chapter 4.

Then, the Hanoi benchmark WDN is presented, where the leak localization and sensor placement techniques are illustrated due to their small size. Then, the Limassol network, where some of the proposed sensor placement techniques presented in Chapter 7, is presented. Finally, the Nova Icària DMA and the Pavones DMA where real leak cases are used to evaluate the proposed leak localization techniques presented in Chapter 5 and Chapter 6.

3.1. Barcelona DMAs

Flow measurements from the inlet of several DMAs from the Barcelona WDN are recorded in two sequences of measurements. Notice that this is a raw data and in consequence there are outliers and missing data intervals.

On the one hand, the first sequence of measurements starts at the 1^{st} of January of 2013 until the 18^{th} of May of 2013, which are 132 complete days. This sequence includes the Bellamar DMA, the Gavà centre DMA, the Parc de la Muntanyeta DMA and the Can Roca DMA. On the other hand, the second sequence of measurements

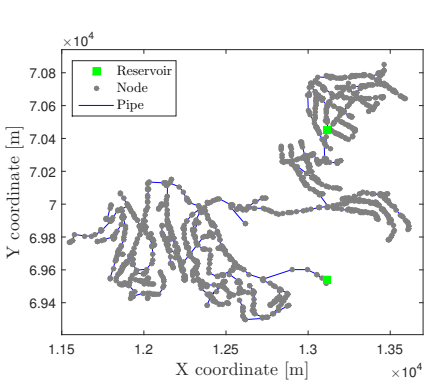
starts at the 31st of August of 2013 until the 3th of March of 2014, which are 185 complete days. This last sequence includes the Gavà centre DMA, the Parc de la Muntanyeta DMA and the Canyars DMA. The sampling rate in all of these DMAs is ten minutes.

Some of these DMAs have more than one reservoir, thus multiple flow measurements are taken. Here, the result of the addition of these flows is depicted for each DMA. All of these DMAs networks are placed next to or near the coast so they are, in general, quite flat in terms of elevation.

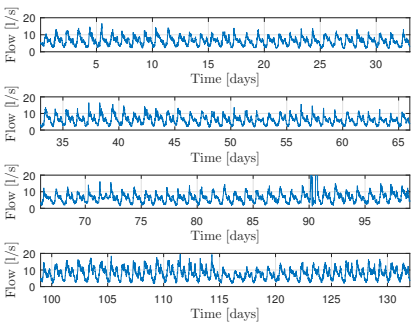
3.1.1. Bellamar DMA

The Bellamar DMA is a large network with two reservoirs without any PRV to control the pressure. This network has a total number of 1523 nodes and a total number of 1544 pipes. The topology of the network is depicted in Figure 3.1a.

This network has one sequence of flow measurements at the inlet with a total water consumption that varies from 2 to 18 [l/s] approximately. This sequence of measurements is shown in Figure 3.1b.



(a) Bellamar DMA network.

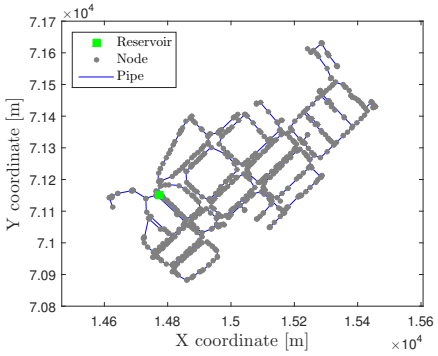


(b) Flow measurements at the Bellamar DMA.

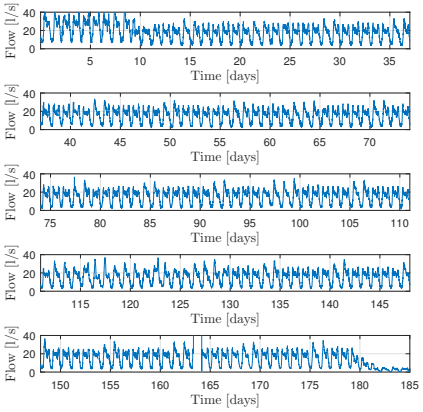
Figure 3.1: Bellamar DMA network and flow measurements.

3.1.2. Canyars DMA

The Canyars DMA is a medium size network composed by one reservoir without PRV, 692 consumer nodes and 717 pipes. The network is depicted in Figure 3.2a. The total water consumption flow in the sequence of measurements varies from 5 and 40 [l/s] approximately and it is shown in Figure 3.2b.



(a) Canyars DMA network.

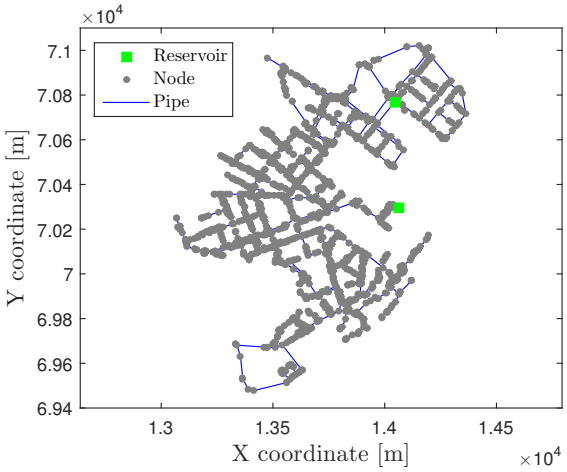


(b) Flow measurements at the Canyars DMA.

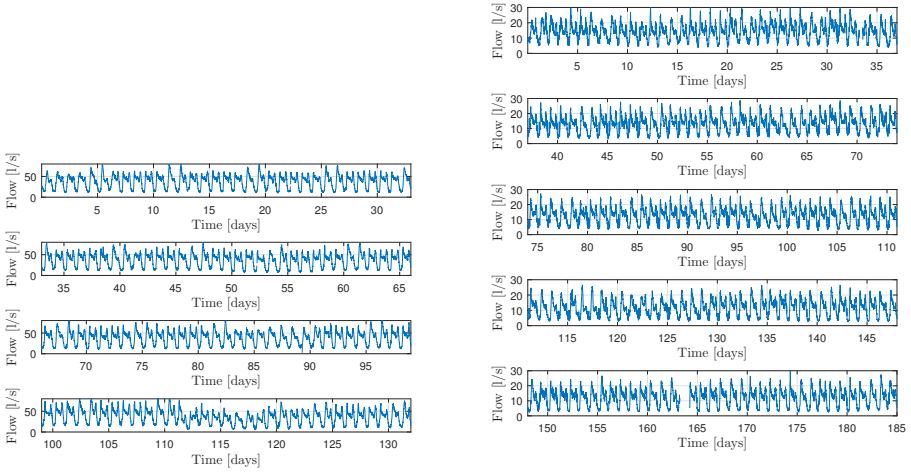
Figure 3.2: Canyars DMA network and flow measurements.

3.1.3. Parc de la Muntanyeta DMA

The Parc de la Muntanyeta DMA is a large network as the Bellamar DMA with two reservoirs without PRVs, 1507 consumer nodes and 1553 pipes. The network is depicted in Figure 3.3a. This network has two sequences of measurements. In the first one, the total water consumption varies from 10 to 70 [l/s] approximately. This flow is shown in Figure 3.3b. The second sequence of flow measurements varies from 8 to 30 [l/s] approximately and is shown in Figure 3.3c.



(a) Parc de la Muntanyeta DMA network.



(b) Flow measurements of the first sequence at the Parc de la Muntanyeta DMA.

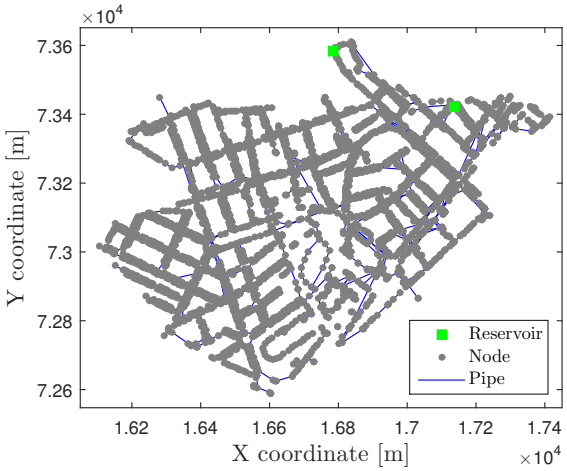
(c) Flow measurements of the second sequence at the Parc de la Muntanyeta DMA.

Figure 3.3: Parc de la Muntanyeta DMA network and flow measurements.

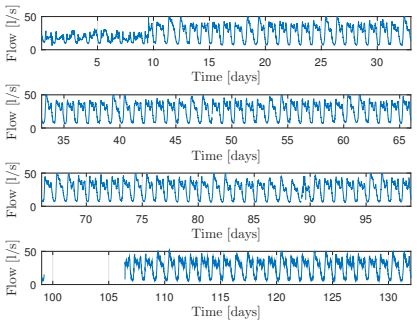
3.1.4. Gavà Centre DMA

The Gavà Centre DMA is a very large network formed by two reservoirs without PRVs, 3373 consumer nodes and 3482 pipes as depicted in Figure 3.4a. This network has two sequences of measurements. In the first one, the total water consumption varies from 8 to 50 [l/s] approximately. This flow is shown in Figure 3.4b. The

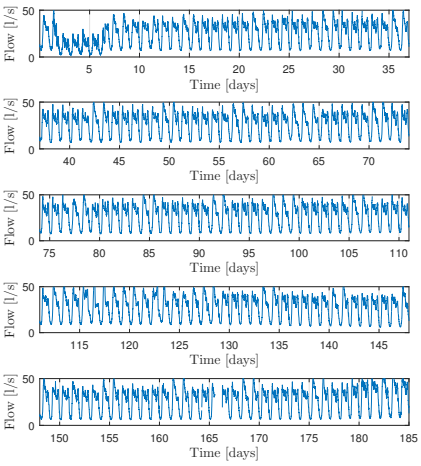
second sequence of flow measurements varies from 5 to 50 [l/s] approximately and is shown in Figure 3.4c.



(a) Gavà Centre DMA network.



(b) Flow measurements of the first sequence at the Gavà Centre DMA.



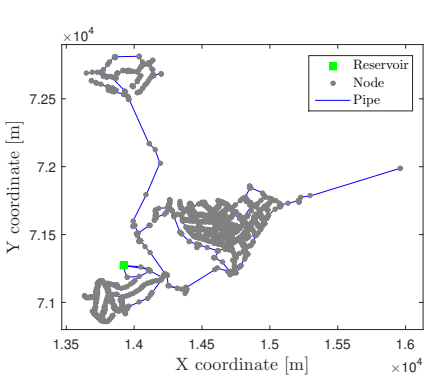
(c) Flow measurements of the second sequence at the Gavà Centre DMA.

Figure 3.4: Gavà Centre DMA network and flow measurements.

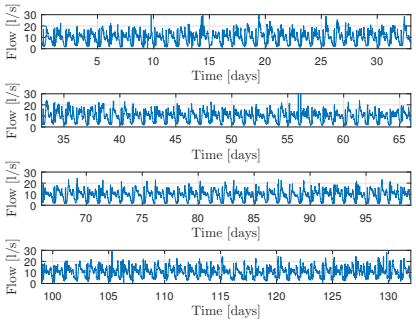
3.1.5. Can Roca DMA

The Can Roca DMA has a total number of 1427 consumer nodes, 1473 pipes and the DMA is feed through an unique reservoir. The topology of the network is depicted in Figure 3.5a. The total flow consumption of this network varies from 2 to 25 [l/s]

approximately. This sequence is plotted in Figure 3.5b.



(a) Can Roca DMA network.



(b) Flow measurements at the Can Roca DMA.

Figure 3.5: Can Roca DMA network and flow measurements.

3.2. Hanoi WDN

The Hanoi WDN is a simplified version of the real WDN placed in Hanoi, Vietnam. This network is one of the benchmarks available in the Epanet 2 software (Rossman, 2000). This WDN consists of one reservoir, 34 pipes and 31 nodes as depicted in Figure 3.6. No PRV is placed inside the network. The Hanoi WDN benchmark has set a fixed nodal demand pattern distribution and a fixed total water consumption at 2991.1 [l/s].

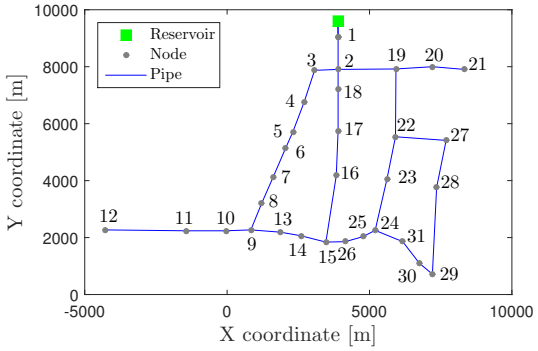


Figure 3.6: Hanoi WDN.

3.3. Limassol DMA

The Limassol (Cyprus) DMA network, presented in Figure 3.7, is a real network with a medium size and consists in one reservoir, 197 consumer nodes and 236 pipes. No PRV is placed in the network. The network has a fixed demand pattern obtained form billing records and a fixed total water consumption of 492.2 [l/s].

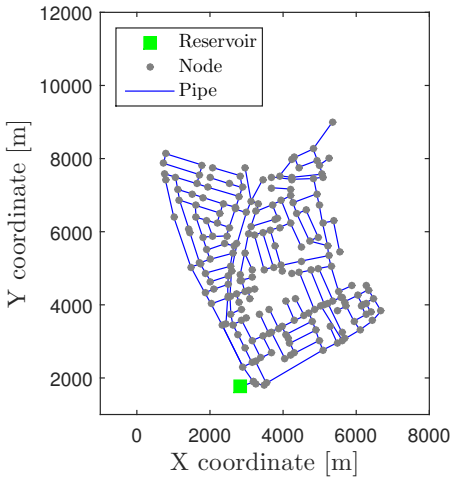


Figure 3.7: Limassol DMA network.

3.4. Nova Icària DMA

The Nova Icària network is one of the DMAs of the Barcelona WDN. This network consists of 3377 nodes, 3442 pipes, two reservoirs and two Pressure Reducing Valves (PRVs), each one located after the reservoirs with the aim of maintaining a certain pressure control level. The number of nodes and pipes of this network is reduced by means of the skeletonization process described in (Pérez and Sanz, 2017) resulting in a network with 1520 nodes and 1664 pipes depicted in Figure 3.8. Measurements of five pressure sensors (the pressure transducers used are the IMP-S-004-010S model[1]

[1] http://www.impress-sensors.co.uk/products/sensor-products/pressure-measurement/industrial-pressure-transducers-transmitters/standard-range-pressure-transmitter/imp-industrial-pressure-transmitter.html

with a resolution of 0.1 [m]) installed in nodes 3, 4, 5, 6 and 7 (highlighted in Figure 3.8), measurement of the flow of the network entering the DMA and the set points for the PRVs are available every 10 minutes using the logger MultiLogS GSM/SMS[2] device. The nodal demand pattern distribution is obtained using billing records.

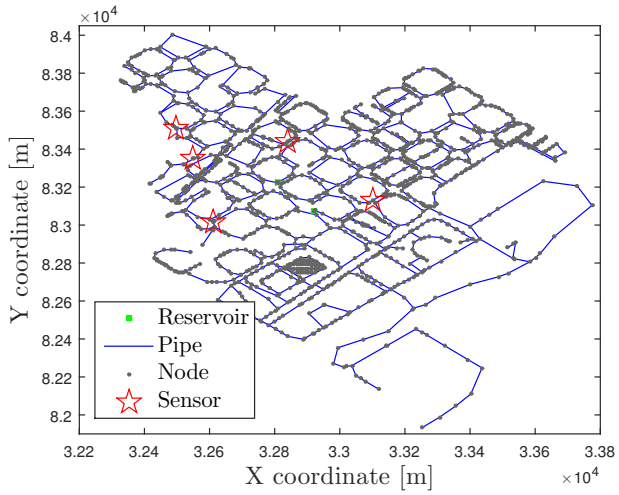


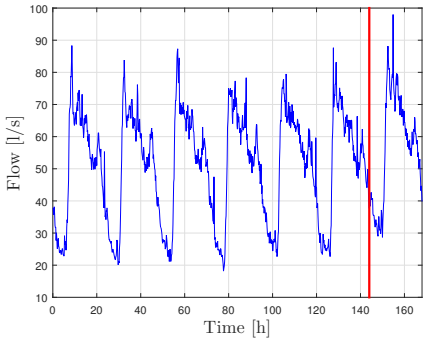
Figure 3.8: Nova Icària DMA network.

3.4.1. Real Case

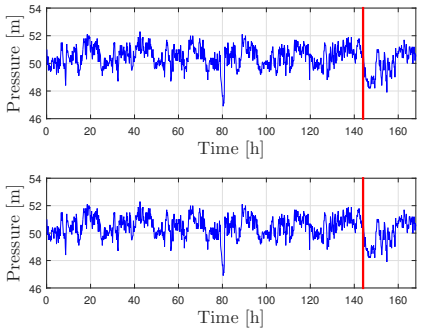
For this case study, real data provided by the water company both under normal operation conditions and under the presence of a real leak have been used. The leak was created by the water company that operates the network by opening a fire hydrant. The experiment took place on December 20, 2012 at 00:30 hour and lasted around 30 hours with a leak size about 5.6 [l/s], being the total demand of water in the range between 23.5 and 78 [l/s] approximately. Moreover, real sensor data for the network in a normal operation scenario of five days before the leak scenario occurred. The relevant data used to perform the leak localization is shown in different figures: Figure 3.9a shows the DMA input flow; Figure 3.9b show the pressure references for the two PTVs; and, finally, Figure 3.9c shows the measurements provided by the five

[2]http://hinco.com.au/shop/type/data-loggers/multilog/

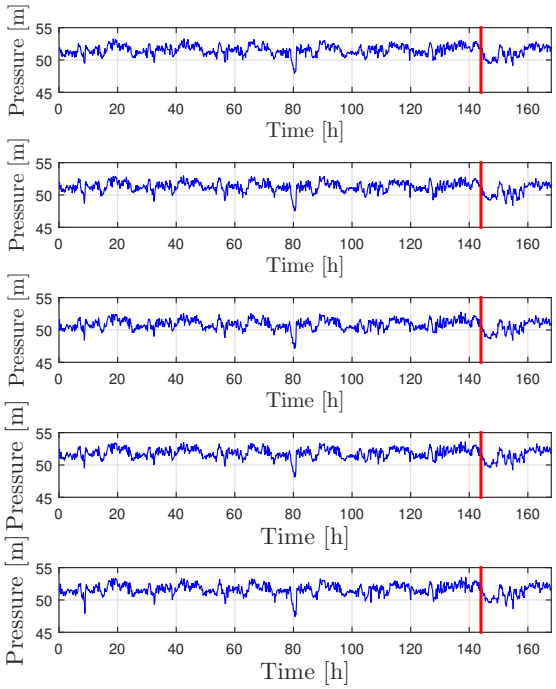
internal pressure sensors. In all these figures, the red line indicates the time instant where the leak is introduced. Finally, an accurate Epanet model of the Network and node demand estimations was provided as well.



(a) Nova Icària flow measurements under nominal conditions (before red line) and faulty conditions (after red line).



(b) Nova Icària PRVs set point values under nominal conditions (before red line) and faulty conditions (after red line).



(c) Nova Icària pressure measurements under nominal conditions (before red line) and faulty conditions (after red line).

Figure 3.9: Nova Icària real case measurements.

3.5. Pavones DMA

Pavones DMA is part of Madrid WDN. Its topology is depicted in Figure 3.10, formed by one reservoir, 608 demand consumer nodes and 638 pipes. The number of pressure inner sensors is ten, and are placed as depicted in Figure 3.10, with sampling rate of two minutes, which is the same for the pressure and flow sensors placed at the inlet.

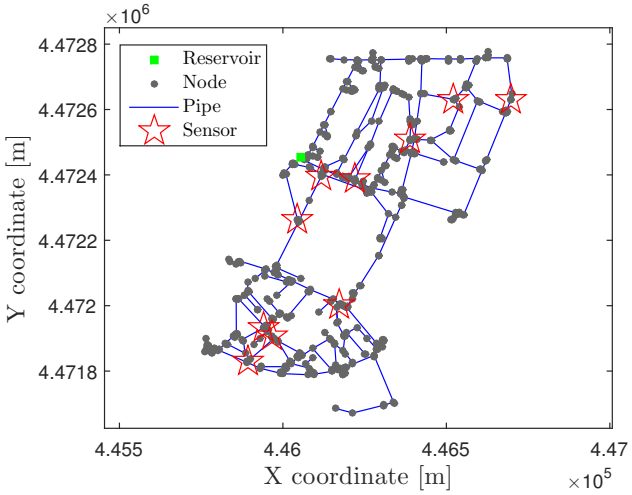
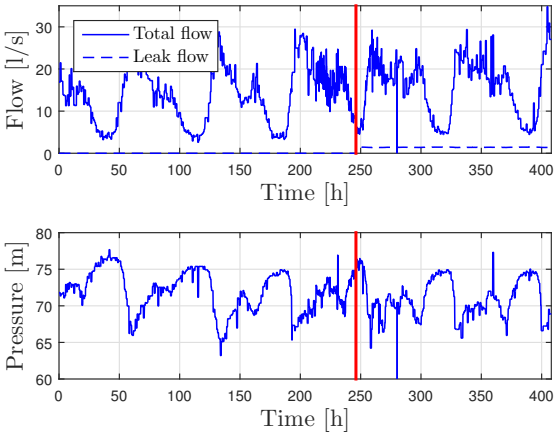


Figure 3.10: Pavones DMA network and their sensor placement.

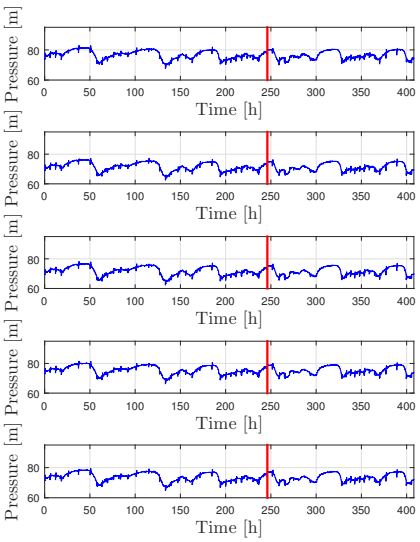
3.5.1. Real Case

Data under nominal conditions is recorded from the 25^{th} of November of 2016 at 03:00 pm to the 29^{th} of November of 2016 ta 00:58 am. A leak artificially created using a fire hydrant with an approximated size of 1.4 [l/s]. Data from this experiment is recorded starting the 29^{th} of November at 04:00 am until the 1^{st} of December of 2016 at 09:58 am, which is 54 hours of data under leaky conditions.

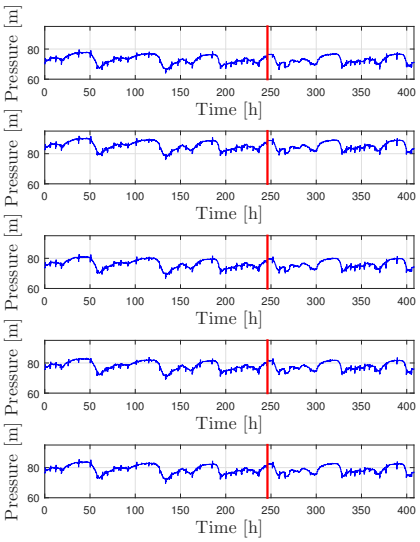
The measurements of flow and pressure at the inlet under nominal and leaky conditions and also the total leak flow are depicted in Figure 3.11a. The ten pressure measurements inside the network under nominal and leaky conditions are depicted in Figure 3.11b and Figure 3.11c.



(a) Measurements at the Pavones DMA inlet under nominal conditions (before red line) and faulty conditions (after red line).



(b) Pressure measurements (part I) in the Pavones DMA under nominal conditions (before red line) and faulty conditions (after red line).



(c) Pressure measurements (part II) in the Pavones DMA under nominal conditions (before red line) and faulty conditions (after red line).

Figure 3.11: Pavones DMA real case measurements.

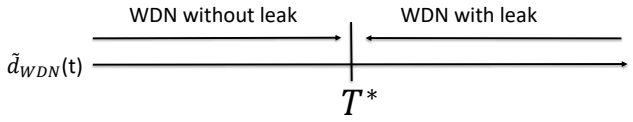
4. Leak Detection

In this chapter, a leak detection technique is presented using the ICI CDT technique presented in Section 2.4 with the purpose of learning from the nominal conditions, specially about the variability of the consumption profile, for determining if the new measurements can be considered as nominal or not.

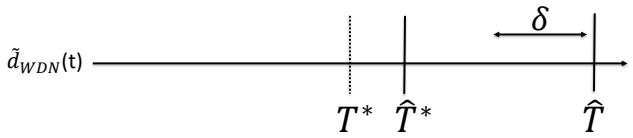
Usually, the WDNs or DMAs inlets (normally corresponding to the reservoirs) are monitored by means of a flow sensor and a pressure sensor. The latter can be substituted by a PRV whose pressure set-point can be considered as the pressure at that point. A common approach when the leak detection problem is addressed is to analyze the current set of measurements coming from the total water consumption $\tilde{d}_{WDN}$ and compare them with the past values obtained in normal conditions (or operations) or by predictions made by a data model. Then, the differences in the comparison are analyzed to see if there is enough evidence to consider a leak inside the network.

Some of these techniques are proposed to be applied only in the period of time known as the Minimum Night Flow (MNF), since is the time region where the water consumption by users is in their minimum value and in consequence, the part of the consumption due to the leak is in the maximum percentage of the total water entering into the network.

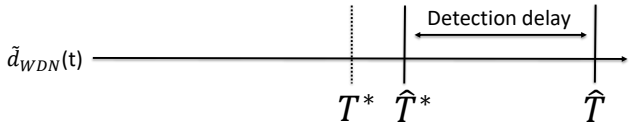
Due to population habits, the profile of the total water consumption presents repetitive patterns on daily basis, that can be altered by the change of habits in weekends, holidays, weather or leaks. Leaks appear as a persistent alteration in the consumption pattern by increasing the consumption of water, which can be detected by



(a) True leak starting time.



(b) Estimated leak starting time.



(c) Leak detection time.

Figure 4.1: Leak detection and starting time estimation scheme.

monitoring the inlet flow. Let us define T^* as the time instant when the leak starts (depicted in Figure 4.1a). The goal of the leak detection technique is to detect the leak as soon as possible. $\widehat{T}^*$ is an estimation of the time instant where the change (leak) has been produced (depicted in Figure 4.1b) and $\widehat{T}$ the moment where the leak has been detected (see Figure 4.1c). Let us also consider that the real leak has a size l, and the estimated leak size of $\hat{l}$, both in [l/s]. So, we define the unknown distributions before and during the leak as

$$\tilde{d}_{WDN}(t) \sim \begin{cases} \Psi(t) & t < T^* \\ \Psi(t) + l & t \geq T^* \end{cases} \tag{4.1}$$

where Ψ is the distribution in nominal conditions and $\Psi + l$ the new distribution due to the existence of a leak inside the network.

It is a common assumption to consider that only one leak can occur at a time (Pérez et al., 2011) being also considered in the proposed leak detection technique.

4.1. Leak Detection Scheme

The problem of leak detection, leak starting time estimation and leak size estimation is addressed as a change point detection problem by means of a sequential monitoring technique. In particular, we design a two layered scheme to deal with leaks in WDNs, which is applied to the total water consumption of the network $\tilde{d}_{WDN}$.
The scheme of the two layered leak detection technique is depicted in Figure 4.2.

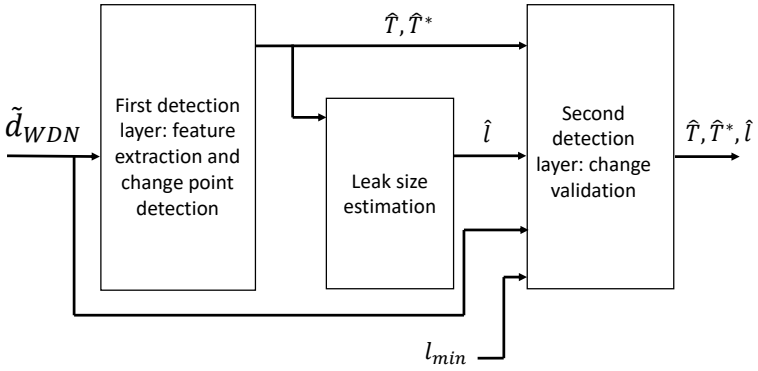


Figure 4.2: Proposed methodology.

4.2. First Layer: Detection

The application of a change detection test provides the detection of an abrupt change in a time series, which in this application is the leak, and the estimated leak starting time as the estimated time where the change has started in the time series.
Change detection methods are typically meant for data streams that are composed of i.i.d. (independent and identically distributed) realizations. This is not the case of the $\tilde{d}_{WDN}$ signal analyzed here. As stated in the previous section, the total flow consumption pattern has a daily repetitive pattern, which is exploited here in conjunction with the self-similarity (Boracchi and Roveri, 2014) on the observation of $\tilde{d}_{WDN}$ to produce a feature that can be well approximated as an i.i.d. realization.

When the system is not in nominal conditions, for example when a leak occurs in the WDN, then the distribution of the measurements changes (notice that the distribution is unknown in nominal and not nominal conditions) presenting then a new, and in consequence, different distribution, as described in (4.1). It is on this change of distribution, that change point methods are designed to work with. But not only leaks can produce this change of distributions, also abnormal consumption or seasonal drifts can produce it, so further analysis over the cause of detection must be done. So, in the first layer the features are extracted and monitored by a sequential monitoring technique to detect subtle changes in the consumption patterns that may be indicative of detected leaks, and the second layer is used to check if the change features the characteristics expected by a change produced by a leak.

From the daily repetitive pattern, only the most stable consumption region, i.e., the part of the daily repetitive pattern where there is less change from day to day due to abnormal consumption is used to apply the proposed two layered leak detection scheme to infer whether a leak in the WDN exists or not.

The Self-Similarity (SS) (Boracchi and Roveri, 2014) approach is used to extract the features to be monitored by the change detection test, in this case, the intersection-of-confidence-interval CDT is used (Basseville and Nikiforov, 1993). The ICI CDT generates an interval of confidence using the past data, the training data set, and a new segment of data along with a user defined parameter Υ to determine the confidence. If the new interval does not intersect with at least one of the previous intervals, a change is detected. This technique is introduced in Section 2.4.

The self-similarity compares a current patch of measurements with the ones in the same position in the reference patterns (as it has been said before, in WDNs every day the same pattern of water consumption is repeated) stored in a training data set recorded under normal, leak-free, conditions (the training data set is called $\mathbf{T}$ and has size m_T), where the most similar patch, according to the ℓ_2 norm, in the training data set is selected, and the difference between centers is computed to create the features. Let us define a patch $\mathbf{s}(t)$ of the actual flow data stream to be monitored

as

$$\mathbf{s}(t) = \{\tilde{d}_{WDN}(t-\zeta), \ldots, \tilde{d}_{WDN}(t), \ldots, \tilde{d}_{WDN}(t+\zeta)\} \tag{4.2}$$

where, ζ is the number of samples taken at each side of the patch center to create the patch and $2\zeta + 1$ is the patch size.

This patch $\mathbf{s}(t)$ is compared with others patches from the same position in the pattern from the self-similarity training data set, where the most similar patch $\pi(t)$ inside the training data set is selected to compute the features. The most similar patch is then calculated as

$$\boldsymbol{\pi}(t) = \underset{\xi}{\operatorname{argmin}} \|\mathbf{s}(t) - \mathbf{s}(\xi)\|_2 \tag{4.3}$$

for $\xi = \{h, \beta + h, \ldots, m_T - \beta + h\}$, where h is the position of the time instant t (the center of the patch) inside the period of the repetitive pattern, β the size of the repetitive pattern and $\|\cdot\|_2$ the ℓ_2 norm computed as

$$\|\mathbf{s}(t) - \mathbf{s}(\xi)\|_2 = \sqrt{\sum_{i=-\zeta}^{\zeta} \left(\tilde{d}_{WDN}(t+i) - \tilde{d}_{WDN}(\xi+i)\right)^2} \tag{4.4}$$

We define $\varrho(t)$ as the distance between patch centers (i.e., features to monitor) at time instant t. This feature $\varrho(t)$ is computed as

$$\varrho(t) = s(t) - \pi(t) \tag{4.5}$$

where $s(t)$ is the value of the center of the vector $\mathbf{s}(t)$ and $\pi(t)$ the value of the center of the most similar patch $\boldsymbol{\pi}(t)$ in the training data set $\mathbf{T}$.

Then, reformulating the presented ICI CDT in Section 2.4 to the current self-similarity features we have the mean from the training data set $\mathbf{T}$ computation as

$$\mu_T = \sum_{t=1}^{m_T} \frac{\varrho(t)}{m_T} \tag{4.6}$$

and the standard deviation as

$$\sigma_T = \sqrt{\sum_{t=1}^{m_T} \frac{(\varrho(t) - \mu_T)^2}{m_T - 1}} \tag{4.7}$$

Then, the updated values of μ and σ are recursively calculated as

$$\mu = \frac{(t - \iota)\mu + \sum_{i=1}^{\iota} \varrho(t - \iota + i)}{t} \tag{4.8}$$

and

$$\sigma = \frac{\sigma_T}{\sqrt{t}} \tag{4.9}$$

Then, the upper bound of the interval is

$$\mathcal{I}^{(+)} = \mu + \Upsilon\sigma \tag{4.10}$$

and the lower bound

$$\mathcal{I}^{(-)} = \mu - \Upsilon\sigma \tag{4.11}$$

The intervals-of-confidence share some region with the others when the system is in nominal conditions, in this case, the network is leak-free. But, when a leak is present in the network, the mean changes and the center of the interval changes which leads to the case that one of the intervals does not share any region with at least one of the previous intervals. This trigger the detection of a change in the monitoring features $\varrho(t)$.

4.2.1. Leak Starting Time Estimation

To estimate the time instant $\widehat{T}^*$ in the data stream where the leak has started, the same mechanism as in the first layer is performed, i.e., the ICI CDT, but using a lower confidence value Υ to trigger a detection at the minimum reasonable change in

the time series since the lower Υ used to create the interval makes their amplitude smaller thus, the not intersection of a new interval with any of the previous can happen with a smaller change in the monitoring features.

4.3. Second Layer: Validation

As stated before, apart from leaks several situations can trigger a detection by the CDT in the first layer since unseen patterns for the training data set can appear. To reduce the false positive rate, this second layer assess if the detection has the reasonable evidence to decide whether the change is due to a leak or not, which in the leak case is a reasonable increment of the total flow consumption. This validation is done in the last δ samples before the CDT has detected a change. To do that, the one sided (right side) Wilcoxon's test (Hollander et al., 2013), which is a paired test that works with the median and does not assume equal variances, is performed. The selection of this test is done taking into account two aspects: the use of the median instead of the mean makes the test robust against outliers and the ability of work with sequences of data with different variances.

This test is applied using the vector of differences between the δ measurements before $\widehat{T}$ and the vector of the averaged sequences contained in the SS training data set in the same time instants, which is the way that the leak size is estimated, plus a minimum leak value l_{min} in [l/s] (i.e., a user defined offset). The test statistic is then compared with a user defined parameter ϑ to assess whether there is enough statistical evidence to reject the null hypothesis.

First, we define the vector $\tilde{d}^{(T)}_{WDN}$

$$\tilde{d}^{(T)}_{WDN}(t) = \frac{1}{m_T/\beta} \sum_{i=0}^{m_T/\beta-1} \mathbf{T}(h(t) + i\beta) \tag{4.12}$$

where $\tilde{d}^{(T)}_{WDN}$ is the average daily consumption pattern inside the SS training data set $\mathbf{T}$ and $h(t)$ is the position of the current measurement in the water daily consumption pattern given the time instant t.

4.3.1. Wilcoxon's Test

So, the proposed validation layer for a given minimum leak size of l_{min} and using the the Wilcoxon's test can be formulated as

$$W = \sum_{i=1}^{N_r} \text{sgn}(w_i - l_{min})\mathcal{R}_i \tag{4.13}$$

where W is the sum of signs for the valid values of the analyzed vector $\mathbf{w}$; N_r is the number of valid samples which are the samples that accomplish $\tilde{d}_{WDN}(t) - \tilde{d}^{(T)}_{WDN}(t) - l_{min} \neq 0$; $\mathbf{w}$ is the ranked valid, non-zero, values according to $\mid \tilde{d}_{WDN}(t) - \tilde{d}^{(T)}_{WDN}(t) - l_{min} \mid$ from smaller to larger; and $\mathcal{R}$ is a vector with the weight of each sample according their position in $\mathbf{w}$, where the weight is the position in the ranking, i.e., the first sample has weight one, the second has weight two, and so on. Then, the W value is compared with the reference tables to obtain the test statistic value. So, if the significance is enough then the detection is validated and the leak localization problem is addressed next. If not, then data before the $\widehat{T}$ is discarded, and a new change is searched starting at time $\widehat{T} + 1$.

4.3.2. Leak Size Estimation

To estimate the leak size value, the inlet flow measurements and the leak detection time $\widehat{T}$ are used. After each detection, the difference in the mean values before $\widehat{T}$ for a data interval δ is computed as the estimated leak size using

$$\hat{l} = \frac{1}{\delta} \sum_{t=\widehat{T}-\delta+1}^{\widehat{T}} \left(\tilde{d}_{WDN}(t) - \tilde{d}^{(T)}_{WDN}(t) \right) \tag{4.14}$$

4.4. Case Study

Real flow data recorded under nominal, leak-free, operations from five DMAs from the Barcelona WDN for two periods, the first for the DMAs Bellarmar, Gavà Centre,

Table 4.1: Different artificial leaks injected into the Barcelona DMAs in [l/s].

Network	Small leak	Large leak	Burst
Bellamar	1	2	4
Gavà Centre (set 1)	2	3	5
Can Roca	3	5	8
Parc de la M. (set 1)	1	2	4
Gavà Centre (set 2)	2	3	5
Canyars	2	3	5
Parc de la M. (set 2)	2	3	5

Parc de la Muntanyeta and Can Roca, which starts the 1^{st} of January of 2013 until the 18^{th} of May of 2013. The second period for the DMAs Gavà Centre, Parc de la Muntanyeta and Canyars starting the 31^{st} of August of 2013 and ending the 3^{th} of March of 2014. The characteristics of these DMAs are detailed in Chapter 3. The days with missing values, with outliers (greater than three times the mean) or negative values are removed for all sequences, giving 100 and 96 days for each period.

Three different sequences with 35 days without leak, where 14 of them are used for training purposes, and 20 with leak (starting at day 1, day 16 and day 31), for each DMA sequence of measurements, different leak sizes are introduced (as a constant increment of the consumption) depending on the total consumption of the network. The leak sizes, in [l/s], are summarized in Table 4.1.

The time range in the repetitive pattern where the techniques are applied is from 10 pm until 8 am. The use of a extended area than the MNF (usually between 2 am and 6 am) allows a faster response, and avoid the needs of consider different training data sets for the weekdays and the weekend, or even, between different days of the week (i.e., consider that different patterns of water consumption exists among different days of the week).

The proposed method has been tuned in the following way:

- The first ten days of each sequence are used as the SS training data set $\mathbf{T}$.
- A patch size of 13 samples.

- The following four days are used for the ICI CDT training data set.
- A $\Upsilon = 1.5$ for the creation of the intervals of confidence.
- A $\Upsilon = 1$ for the creation of the intervals of confidence in the leak starting time procedure.
- A window size of $\iota = 20$ samples for the creation of each interval for the ICI CDT.
- A minimum leak size of $l_{min} = 0.5$ [l/s].
- A value of $\vartheta = 0.05$ to reject the null hypothesis.
- A number of $\delta = 36$ samples (six hours) for the Wilcoxon's test and leak size estimation.

To tune the technique, a reasonable number of sequences for the SS training data set is chosen to have a good variability in them; a larger patch size is chosen to catch the most similar trend; the ICI CDT data set length is selected to avoid tardiness response; the both confidence values are set low since the validation layer (for the detection) prevent the most false positives made in the first layer and allow a faster detection; a window size big enough not to have outliers that can lead to a false detection; the minimum leak size is made according to expected minimum detectable leak size; the threshold to reject the null hypothesis is chosen bigger than zero to minimize false rejections and the number of samples chosen to validate and estimate the leak size has been empirically chosen.

Three other approaches already published have also been tested into these data sets in order to compare the performance of the proposed technique. In order to have a comparison in similar conditions, the methods are tuned to have the same false positive rate.

The first method is the one in which the proposed method is based (Boracchi and Roveri, 2014). However, here it is used in the detection layer of the proposed method. It is tuned with a $\Upsilon = 4.6$ (to have the same FPR as the one obtained with the proposed technique), while the remaining tuning is the same as the proposed approach.

Another approach (Palau et al., 2012) uses historical flow measurements (in this case the training data sets used for the SS and ICI CDT) to built a PCA model where the samples correspond to each data interval of measurements considered and the attributes are the samples in that data interval. The data is centered to zero and normalized using the mean and the standard deviation respectively for each attribute. Then, the PCA model is constructed and the load of new measurements according to the principal components (i.e., the number of principal components are set to contain at least 95 % of the total information of the original representation) are analyzed by an statistic. In (Palau et al., 2012), different PCA are constructed depending on the time of the day, one for the night, one for the morning and one for the afternoon, and then, these three PCA are built for the weekdays and the weekend. Here, only one PCA has been built since the time range used is quite stable and the difference between weekdays and weekends with respect to different days in the week is not noticeable. To decide the threshold that indicates if leaks exists or not, the mean value of the load in the training data set plus a 3.7 times the standard deviation is used in order to have the same FPR than the proposed approach.

Finally, in (Ye and Fenner, 2011) an approach based on a set of adaptive Kalman filters is used. An adaptive Kalman filter for each sample measurement inside a week is used to predict the flow and generate normalized residuals (i.e., the residual is divided by the actual measurement) with the actual measured flow. The vector of normalized residuals is then averaged with a moving average of one week, and the results are compared with a threshold. If the value overpass the threshold, the detection is made, and the leak size is estimated using the actual measurement and the normalized residual when the detection is made. Rather than a week period, here, the time range between 10 pm to 8 am is used, so only 48 adaptive Kalman filters are used. The threshold used is 0.19 and the initial values are the ones proposed in the paper.

The results for the four methods for the small leaks case are summarized in Table 4.2, for the large leaks in Table 4.3 and for the bursts in Table 4.4. The indicators used

are the ones detailed in Section 2.3.

Table 4.2: Comparison of leak detection performance with small leaks.

Method	FPR [%]	FNR [%]	DD [h]	DTD [h]	Δl [l]
Proposed	9.5	9.5	123.9	76.4	0.8
ICI CDT	9.5	57.1	221.7	195.6	-
PCA	9.5	33.3	124.7	-	-
Kalman	9.5	47.6	146.0	-	4.2

Table 4.3: Comparison of leak detection performance with large leaks.

Method	FPR [%]	FNR [%]	DD [h]	DTD [h]	Δl [l]
Proposed	9.5	4.8	73.9	28.6	0.8
ICI CDT	9.5	23.8	268.6	221.4	-
PCA	9.5	23.8	73.7	-	-
Kalman	9.5	4.8	29.4	-	2.6

Table 4.4: Comparison of leak detection performance with bursts.

Method	FPR [%]	FNR [%]	DD [h]	DTD [h]	Δl [l]
Proposed	9.5	0	49.2	3.9	1.2
ICI CDT	9.5	0	122.7	72.1	-
PCA	9.5	4.8	48.0	-	-
Kalman	9.5	0	25.5	-	2.2

As it can be seen from these tables, all methods struggle to deal with small leaks, which in some networks can be confused by abnormal consumptions. Regarding to the proposed method, and in extension, also to the ICI CDT method, the problem of flow sensor resolution is added since in the creation of the features by the self-similarity there is not enough granularity in some cases. This can be seen in Figure 4.3 where variations of water consumption are not well represented due to the low resolution. Moreover, in the the small leak the increment due to the leak is not noticeable.

The proposed method improves with respect to the one in which is based (Boracchi and Roveri, 2014), apart from adding the ability of estimating the leak size. In particular, in all areas except of the estimated leak starting time which is performing similarly, the the approach proposed in this chapter outperforms the detection results

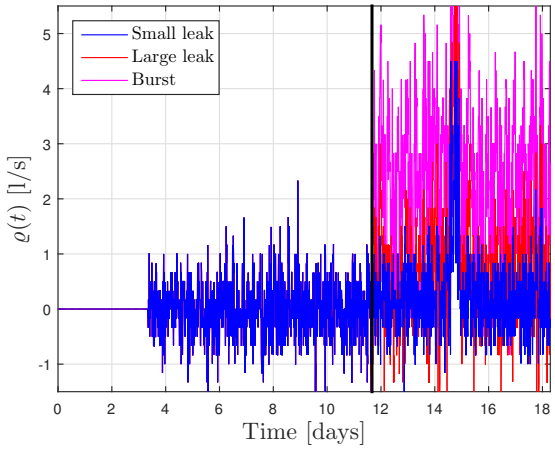


Figure 4.3: Self-similarity ϱ feature in Ballamar DMA for the three leak cases, the vertical black line marks the leak starting time.

compared to the one proposed in (Boracchi and Roveri, 2014).

Regarding to the other two methods, one the one hand, the proposed approach presents a more robust behavior since although provides the same FPR, the FNR is better, specially compared to the PCA method. One the other hand, the proposed approach performs better with small leaks than the adaptive Kalman filter technique. The drawback is that is slower than the method based on the adaptive Kalman filters. The proposed method provides more information than the others, specially compared with the PCA since it only provides the leak detection. In the case of the method based on the adaptive Kalman filters, which also provides a leak size estimation, this is far less accurate that the one provided by the proposed method. From these results, we can state that the proposed method is the most robust but has the drawback of being slower when performing the detection.

5. Model-Based Leak Localization

In this chapter, a new model-based approach for leak localization in WDNs using pressure models and classifiers is presented. This methodology is intended to be used after the leak has been detected by means of the analysis of the night DMA water demands (Puust et al., 2010) or the method proposed in Chapter 4, and after the application of the validation and reconstruction methodology described by (Cugueró-Escofet et al., 2016) to the sensors used for leak localization. Following a model-based methodology successfully tested in (Pérez et al., 2011) and (Pérez et al., 2014), a pressure model of the considered WDN is used in a first stage to compute residuals that are indicative of leaks. In a second stage, a classifier is applied to the obtained residuals with the aim of determining the leak location. In particular, the k-NN classifier and the multi-class Bayesian classifier presented in Section 2.4 are proposed and tested in this chapter. This on-line scheme relies on a previous off-line work in which the hydraulic model is obtained and the classifier is trained with data generated by extensive simulations of the hydraulic model. These simulations consider three types of uncertainties: leaks with different magnitudes in all the nodes of the network (as discussed in the Chapter 2, it is a common approach to consider leaks only at nodes), differences between the estimated and real consumer water demands and noise in pressure sensors. The underlying idea is to obtain a classifier able to distinguish the leak location independently of the unknown real leak magnitude and the presence of uncertainties associated to the water demands and the pressure measurements.

5.1. Principle of Model-Based Leak Localization Approaches

Model-based approaches aim to localize leaks in a water distribution network by comparing pressure measurements (pressure measurements are preferred by the companies in charge of water networks because they are easier and cheaper to install and maintain) with their estimations obtained by using the hydraulic network model. Usually, this methodology is used for localizing leaks within a given leak size range defined by the water network management company. The minimum size is related to the sensor resolution and modelling/demand uncertainty, and the maximum size is defined as the value such that the leak behaves as a burst such that it can be seen in the street. Model-based leak localization methods are based on comparing the monitored pressure disturbances caused by the current leak at certain inner nodes of the WDN or DMA with the theoretical pressure disturbances caused by all potential leaks obtained by using its respective model (Pérez et al., 2014). This comparison uses the residual vector $\mathbf{r} \in \mathbb{R}^{n_s}$, obtained from the difference between the measured pressure at DMA inner nodes $\mathbf{p} \in \mathbb{R}^{n_s}$ and the pressure at these nodes calculated by using the network model considering a leak-free scenario $\hat{\mathbf{p}} \in \mathbb{R}^{n_s}$, i.e.

$$\mathbf{r}(t) = \mathbf{p}(t) - \hat{\mathbf{p}}(t) \tag{5.1}$$

The dimension of the residual vector $\mathbf{r}$, n_s, depends on the number of inner pressure sensors installed in the network. In recent years, some optimal sensor placement algorithms have been developed to determine where the pressure sensors should be installed inside the DMA with minimum economical costs (number of sensors), and guaranteeing a suitable performance regarding leak localization, see (Pérez et al., 2011), (Casillas et al., 2013), (Sarrate et al., 2014b) among others.

The number of potential leaks l_i (l is the leak magnitude in [l/s]) with $i \in \mathbb{R}^{n_n}$, is considered to be equal to the number of WDN or DMA nodes n_n, since from the

modeling point of view, as proposed by (Pérez et al., 2011) and (Pérez et al., 2014), leaks are assumed to occur in these locations.

5.2. Limitations of Sensitivity Analysis Approaches

Most model-based leak localization approaches rely on the sensitivity-to-leak analysis (Pérez et al., 2011; Pérez et al., 2014) where the theoretical pressure disturbances caused by all potential leaks are stored in the leak sensitivity matrix $\mathbf{\Omega} \in \mathbb{R}^{n_s \times n_n}$ (with as many rows as DMA inner pressure sensors, n_s, and as many columns as potential leaks in all nodes n_n). Then, leak isolation is based on matching the residual vector (5.1) with the columns of the sensitivity matrix by using some metrics as for example the correlation or the angle (see (Casillas et al., 2012) for details). The leak sensitivity matrix can be mathematically formalized as follows

$$\mathbf{\Omega} = \begin{pmatrix} \frac{\partial r_1}{\partial l_1} & \cdots & \frac{\partial r_1}{\partial l_{n_n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_{n_s}}{\partial l_1} & \cdots & \frac{\partial r_{n_s}}{\partial l_{n_n}} \end{pmatrix} \tag{5.2}$$

where each element $\Omega_{i,j}$ measures the effect of the leak l_j in the residual r_i associated to the pressure at node i. In practice, it is extremely difficult to calculate $\mathbf{\Omega}$ analytically because a water distribution network is a large scale multivariable non-linear system which equations can only be solved numerically. Thereby, the sensitivity matrix is generated by simulation of the network model and evaluating the sensitivity $\Omega_{i,j}$ as

$$\Omega_{i,j} = \frac{\hat{p}_i^{(l_j^{(0)},0)} - \hat{p}_i^{(0)}}{l_j^{(0)}} \tag{5.3}$$

where the superscript ^{0} denotes the absence of noise in measurements and nodal demand uncertainty, $\hat{p}_i^{(l_j^{(0)},0)}$ is the predicted pressure in the node when a nominal

(fixed value) leak $l_j^{(0)}$ is injected in node j and $\hat{p}_i^{(0)}$ is the predicted pressure associated with the sensor i under a scenario free of leaks (Pérez et al., 2011). The superscript $^{(0)}$ denotes the nominal conditions (i.e., the estimated demand $\hat{\mathbf{d}}$ and without artificial noise added). Then, the sensitivity matrix is obtained by repeating this process for all n_n potential leaks.

Different techniques and metrics are used to exploit the information contained in (5.2). First, in (Pérez et al., 2011), the sensitivity matrix (5.2) is binarized using a threshold with the aim, if it is possible, to have a different signature for each potential leak location. Then, leak localization is based on matching the current binarized residual with the columns of (5.2) and selecting the node or set of nodes with lowest Hamming distance. Later, in (Quevedo et al., 2012), the sensitivity matrix (5.2) is not binarized and the Pearson's correlation coefficient between the current residual and the columns of the (5.2) is computed as

$$\rho_{\mathbf{r},\mathbf{s}^{(l_i)}} = \frac{\text{cov}(\mathbf{r}, \mathbf{s}^{(l_i)})}{\sqrt{\text{cov}(\mathbf{r}, \mathbf{r})\text{cov}(\mathbf{s}^{(l_i)}, \mathbf{s}^{(l_i)})}} \tag{5.4}$$

where $\mathbf{r}$ id the vector of actual residuals to analyze, $\mathbf{s}^{(l_i)}$ is the i^{th} column of (5.2) and cov() is function that returns the covariance between the two vectors. The column where the largest correlation is computed is the index of the node candidate. Similar approach is proposed by (Casillas et al., 2012) where a metric angle, the cosine, between the current residual and the columns from (5.2) is obtained as

$$\text{cosine} = \frac{\mathbf{r} * \mathbf{s}^{(l_i)}}{\|\mathbf{r}\|_2 \|\mathbf{s}^{(l_i)}\|_2} \tag{5.5}$$

The leak (column) with a value (5.5) closest to one is the node candidate.

An important drawback of the leak sensitivity approach is that the practical evaluation of (5.3) depends on the nominal leak l_j (Blesa et al., 2012, 2016). If the real leak size is different from the nominal one, the real sensitivity will be different from the one computed using (5.3). Moreover, the sensitivity is also affected by the nodal demand uncertainty (Cugueró-Escofet et al., 2015b) since this demand is not measured but estimated using historical records of water consumption and

using the aggregated DMA consumption pattern. These uncertainties, besides noise in the measurements, will deteriorate the leak localization results obtained by using the sensitivity approach. The approach proposed in this chapter aims to overcome these difficulties.

5.3. Basic Architecture and Operation

The method for on-line leak localization proposed in this chapter relies on the scheme depicted in Figure 5.1, and it is based on computing pressure residuals and analyzing them with a classifier. The hydraulic model is built using the Epanet hydraulic simulator (Rossman, 2000) by considering the DMA structure (pipes, nodes and valves) and network parameters (pipe coefficients). After the corresponding calibration process using real data, it is assumed that the hydraulic model is able to represent precisely the WDN behavior. However, it must be noted that the model is fed with estimated water demands in the nodes $(\hat{d}_1, \ldots, \hat{d}_{n_n})$. In practice, nodal demands $(d_1, \ldots, d_{n_n})$ are not measured (except for some particular consumers where Automatic Metering Readers (AMRs) are available) and are typically obtained by the total measured DMA demand $\tilde{d}_{WDN}$ and distributed at nodal level using historical consumption records. Hence, the residuals are not only sensitive to leaks but also to differences between the real demands and their estimated values. Additionally, pressure measurements are subject to the effect of sensor noise ν and this also affects the residuals. Taking all these effects into account, the classifier must be able to localize the real leak present in the WDN, that can be in any node and with any (unknown) magnitude, while being robust to the demand uncertainty and the measurement noise. Finally, the operation of the network is constrained by some boundary conditions $\mathbf{c}$ (such as the position of internal valves, reservoir pressures and flows) that are known (measured) and must be taken into account in the simulation and can also be used as inputs for the classifier.

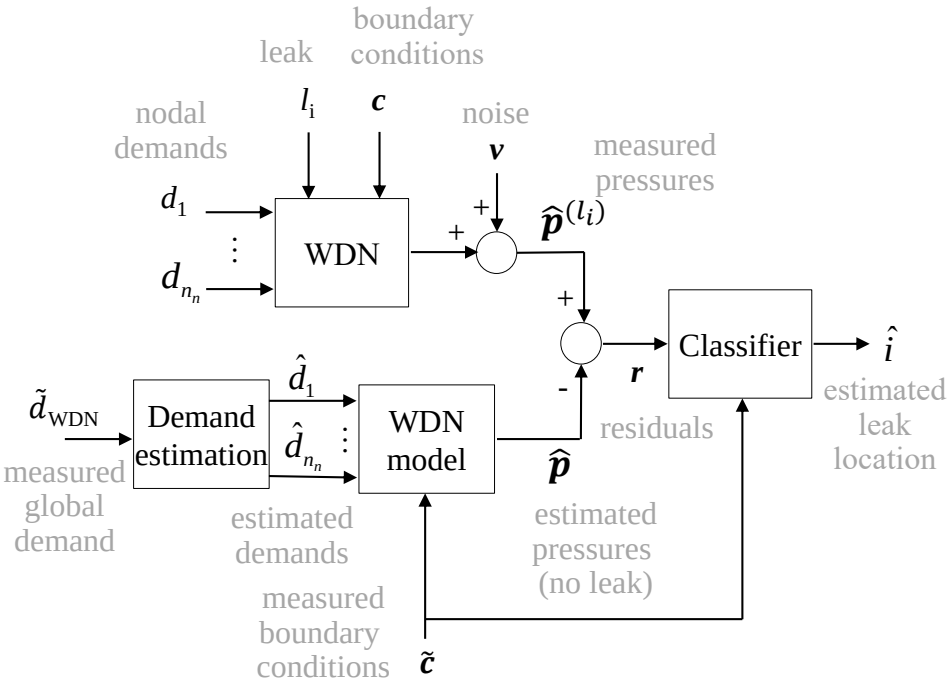


Figure 5.1: Leak localization scheme.

5.4. Methodology Overview

The exploitation of the architecture presented Figure 5.1 relies on a methodology that distinguishes several off-line and on-line procedures.

5.4.1. Off-line Stage

The application of the architecture presented in Figure 5.1 relies on an off-line work whose main goal is to obtain a classifier able to distinguish the potential leaks under the described uncertainty conditions. In particular, the method proposed in this chapter considers an off-line design based on the following stages:

- Modelling - A hydraulic model for the WDN is obtained, calibrated and implemented with Epanet. The model is basically built by taking into account the network structure and by applying flow balance conservation and pressure

loss equations described in Section 2.1.

- Data generation - The model implemented is extensively used to generate data in the residual space for each possible leak and for different operating and uncertainty conditions.

- Classifier training and evaluation - The classifier is first trained with a subset of the initial data set, then it is applied to the testing data in order to estimate its performance.

The data generation stage is critical since the availability of representative data is a necessary condition for obtaining a good classifier. Since the amount of data collected from the real monitored WDN is limited, a way to obtain a complete training data set is by using the hydraulic simulator. Hence, the training and testing data are generated by applying the scheme depicted in Figure 5.2, which is similar to the one presented in Figure 5.1 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 leaks (injected as described in Section 2.1).
The presented data generator scheme is exploited in order to:

- Generate data for all possible leak locations, i.e. for all the different nodes in the WDN ($\bar{l}_i, \quad i = \{1, 2, \ldots, n_n\}$).

- Generate data for each possible leak location with different leak magnitudes within a given range ($\bar{l}_i \in [l_i^{(-)}, l_i^{(+)}]$).

- Generate sequences of demands $\bar{\mathbf{d}}$ and boundary conditions $\bar{\mathbf{c}}$ that correspond to realistic typical daily evolution in each node.

- Simulate differences between the real demands and the estimations computed by the demand estimation module ($(\bar{d}_1, \ldots, \bar{d}_{n_n}) \neq (\hat{d}_1, \ldots, \hat{d}_{n_n})$).

- Take into account the measurement noise in pressure sensors, by generating synthetic noise ($\bar{v}$).

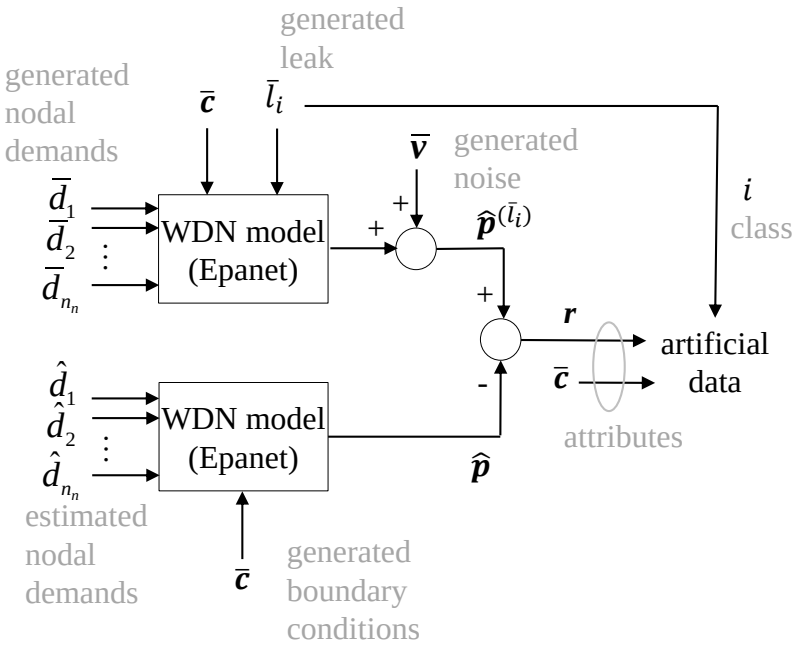


Figure 5.2: Residual data generation scheme.

The artificial data obtained from simulations is divided into training and testing sets. The training stage is based on a learning procedure where the input is the (labeled) training data set and the result is a classifier that must be able to correctly classify new data instances into the correct class. The generalization ability of the obtained classifier is checked in the testing stage, in which the performance indexes are computed for the testing data set.

The details of the training stage are particular of the type of classifier used. The results presented in the following sections have been obtained by using two different well-known classifiers: the k-Nearest Neighbor (k-NN) classifier, which is non-parametric, and the multi-class Bayesian classifier, which is parametric. The details about the training of both classifiers will be provided in the next subsections.

5.4.2. On-line Stage

Once the classifier has been trained and validated, it can be used on-line to localize leaks. According to Section 5.3, the classifier can be directly used, once the leak has been detected, to localize leaks based on the instantaneous values of the computed residuals.

However, this strategy may provide limited results if there is a high level of uncertainty. The use of a temporal reasoning that takes into account not only the instantaneous values of the residuals but all the values within a time horizon is considered as already proposed in (Casillas et al., 2012). This idea is implemented in different forms depending on the type of classifier that is used, details are provided in the next subsections.

5.5. k-NN Classifier Implementation

5.5.1. The k-NN Classifier

The k-NN algorithm presented in Section 2.4 is adapted to the leak localization problem by assuming that each potential leak location represents a different class. Then, the artificial data with the artificial leaks injected generated by a hydraulic simulator is directly stored in the training data set to be compared with the new instances. If some attribute has a different scale (i.e., the magnitude of the variability of each attribute) to the others, it is recommended to scale it to a similar values of the rest of the attributes in order to avoid uncompensated attributes, for example, the total water consumption has a larger variability than the residuals, so the flow must be normalized.

5.5.2. Time Reasoning

If the uncertainty in the demands, the leak magnitude or the noise level are large then the direct application of the classifier can provide poor leak localization results. To reduce the impact of the demand uncertainty, leak magnitude and noise, typically the analysis of the residuals evolution is performed in a time horizon, i.e., the values for the residuals in the last N time instants are considered (as applied in (Casillas et al., 2012)).

As proposed in Section 2.4, a time reasoning is applied here, where the information stored in the confusion matrix $\mathbf{\Gamma}$ generated using a validation data set is used to estimate the probabilities $P(l_i|l_j)$ (i.e., the probabilities that given a diagnosis of leak in node j the leak is in node i). So, the sum of the probabilities is used to infer the diagnosis by taking the node with the largest value as the node candidate in the time horizon $[t - N + 1, t]$.

5.6. Bayesian Classifier Implementation

5.6.1. Bayesian Classification

The multi-class Bayesian classifier is adapted in a similar way as the k-NN case. Each potential leak location is assumed to be a different class for the classifier. Then, we have the following reformulation.

Given the residual vector $\mathbf{r}$, the objective is to apply a Bayesian leak discrimination procedure in order to identify which leak or leaks may occur based on the observed behavior. Such a diagnosis procedure based on Bayesian reasoning explained in Section 2.4 is adapted to the leak localization problem is explained next.

At every time sample t, the probability of a leak occurrence is estimated as a result of the application of the Bayes Rule

$$P(l_i \mid \mathbf{r}(t)) = \frac{P(\mathbf{r}(t) \mid l_i)P(l_i)}{P(\mathbf{r}(t))} \text{ for } i = 1, \ldots, n_n \tag{5.6}$$

where $P(l_i \mid \mathbf{r}(t))$ is the posterior probability that the leak l_i had caused the observed residual vector $\mathbf{r}(t) = (r_1(t), \ldots, r_j(t))^T$, $P(\mathbf{r}(t) \mid l_i)$ is the likelihood of the residual $\mathbf{r}(t)$ assuming that the active leak is l_i, $P(l_i)$ is the prior probability for the leak l_i, and $P(\mathbf{r}(t))$ is a normalizing factor given by the Total Probability Law, which in the proposed leak localization case is

$$P(\mathbf{r}(t)) = \sum_{i=1}^{n_n} P(\mathbf{r}(t) \mid l_i) P(l_i) \tag{5.7}$$

Regarding prior probabilities, unless we have any additional information, an unprejudiced starting point is to consider all the potential leak locations equally probable, that is, $P(l_i) = \frac{1}{n_n}$ for $i = 1, \ldots, n_n$. To estimate the likelihood value $P(\mathbf{r}(t) \mid l_i)$, we need to perform a previous calibration task in order to obtain the joint probability density function for each leak in the residual space, $P(\mathbf{r} \mid l_i)$ for $i = 1, \ldots, n_n$. The calibration stage is detailed in a next section. Note that, in contrast to standard Naïve Bayesian classifiers, we do not need to assume independence between the residuals.

The application of (5.6) produces a set of values $P(l_i \mid \mathbf{r}(t))$, $\sum_{i=1}^{n_n} P(l_i \mid \mathbf{r}(t)) = 1$, that can be used to decide where the leak is located. So, the leak with the highest posterior probability is the node candidate provided by the Bayesian classifier. Another option could be to select a set of node candidates with the posterior probability above a pre-specified threshold.

5.6.2. Recursivity

The results can be improved if (5.6) is recursively applied, that is, if the posterior probability $P(l_i \mid \mathbf{r}(t))$ is used as the prior probability for the next sample time. This way, as long as new measurement data is available, the probabilities are updated and many of the competing leaks can be discarded.

The only drawback is that if any of the leaks take the posterior probability value of 1 at any t, then all the remaining leaks take the 0 probability value, therefore preventing them to have a future value different from zero due to the recursive appli-

cation of (5.6). This drawback can be easily overcomed by forcing all probabilities to have a maximum value of, say, 0.99. When a leak l_i presents the probability $P(l_i \mid \mathbf{r}(t)) > 0.99$, we force it to be $P(l_i \mid \mathbf{r}(t)) = 0.99$ and we can force the remaining leaks to be $P(l_n \mid \mathbf{r}(t)) = \frac{1-0.99}{n_n-1}$ for $n = 1, \ldots, n_n$ and $n \neq i$.

5.6.3. Bayesian Time Reasoning

Additionally, the results can be improved if a time horizon N is introduced. In this case, the posterior probability can be computed on the basis of the N previous time samples, that is, to compute $P(l_i \mid \mathbf{r}(t))$, we recursively can apply the following equation

$$P(l_i \mid \mathbf{r}(t-N+j)) = \frac{P(\mathbf{r}(t-N+j) \mid l_i) P(l_i \mid \mathbf{r}(t-N+j-1))}{P(\mathbf{r}(t-N+j))} \quad \text{for } i = 1, \ldots, n_n \text{ and } j = 1, \ldots, N \tag{5.8}$$

where an unprejudiced starting point may be all the potential leak locations equally probable as $P(l_i \mid \mathbf{r}(t-N)) = \frac{1}{n_n}$ for $i = 1, \ldots, n_n$.

5.6.4. Calibration of the Probability Density Functions

Unlike the k-NN classifier, the Bayes classifier requires a more elaborated training phase where a joint Probability Density Function (PDF) for each leak class in the residual space, $P(\mathbf{r} \mid l_i)$ for $i = 1, \ldots, n_n$, has to be estimated.

The first step is to decide the probability family. The Law of Large Numbers states that most situations lead to a Gaussian probability density function if the number of samples is high enough. Several tests can be applied to the residual values to assess if they are Gaussian distributed or not. For instance, we can apply the well-known one-dimensional Kolmogorov-Smirnov (Daniel et al., 1978) or the Anderson-Darling (Stephens, 1974) tests, among others.

Figure 5.3 shows two leak distributions calibrated by means of Gaussian probability function. Leak 1 is better adjusted because it takes into account the cross-correlation between residuals r_1 and r_2. On the other hand, leak 2 is adjusted by assuming statistic independence between residuals r_1 and r_2 and therefore the fitting is not so accurate. Note also that other probability distribution families different from Gaussian could be used, including multimodal and non-parametric distributions.

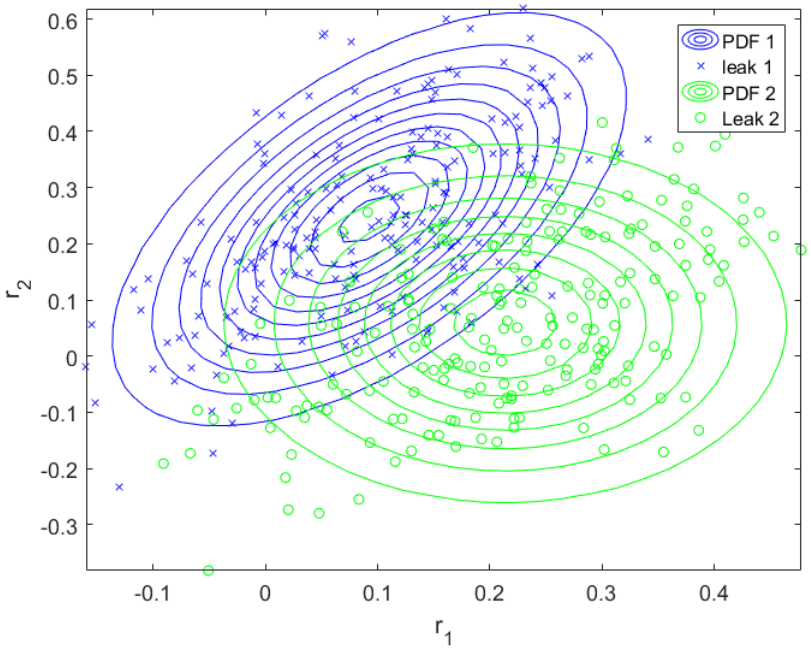


Figure 5.3: PDF calibration for leaks 1 and 2.

5.7. Case Studies

In this section, a simplified WDN and a DMA case studies of increasing size and complexity (Hanoi and Nova Icària) are introduced to assess the performance of the proposed methodology.

As already previously discussed, leaks are considered in any of the demand nodes. The known variables are the input pressures and flows of the networks (reservoir boundary conditions) and some pressures at the inner nodes of the networks where

sensors would be located (see Chapter 7 for details about optimal sensor location). It is considered that the demand pattern is known for all demand nodes but with some uncertainty as proposed by (Cugueró-Escofet et al., 2015b). The leak magnitude is assumed to be unknown but bounded by a known interval (minimum and maximum leak magnitudes). Finally, noise in pressure sensors is considered too.
For the considered networks, leak localization results under different uncertainty scenarios are presented and discussed. Moreover, for the second (and biggest) network the results of localizing a real leak case are also presented.

5.7.1. Hanoi WDN Case Study

The proposed methodology has been first applied to the simplified model of the Hanoi (Vietnam) WDN, depicted in Figure 5.4. Measurements of two inner pressure sensors arbitrary placed in nodes 14 and 30 are available.

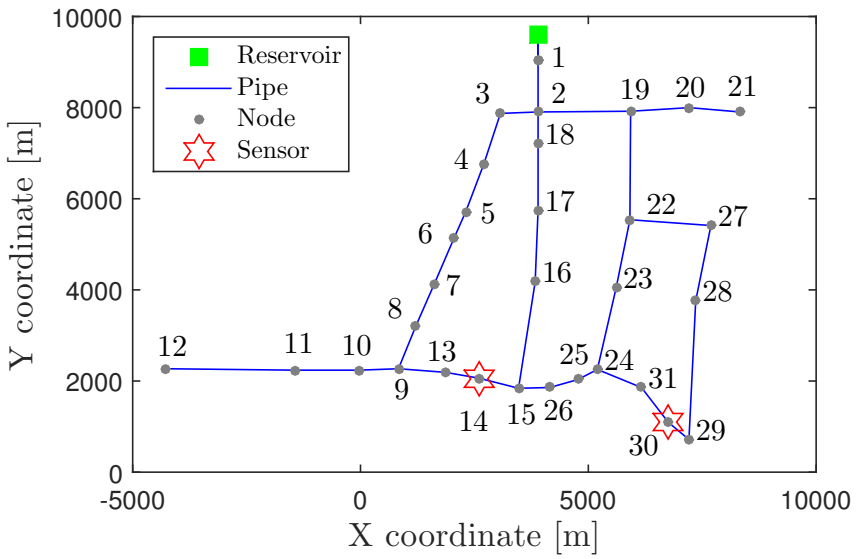


Figure 5.4: Hanoi topological WDN and their sensor placement.

Uncertainty Effect in the Residual Space

Without leaks and in a scenario without uncertainty (i.e., no demand and leak uncertainties, and no noise in measurements), the residuals should be close to zero. However, in case of a leak, and still in a scenario without uncertainty and a fixed total eater consumption of $\tilde{d}_{WDN} = 2991.1$ [l/s], the residuals are points in the residual space as it is shown in Figure 5.5 for a leak size of 50 [l/s].

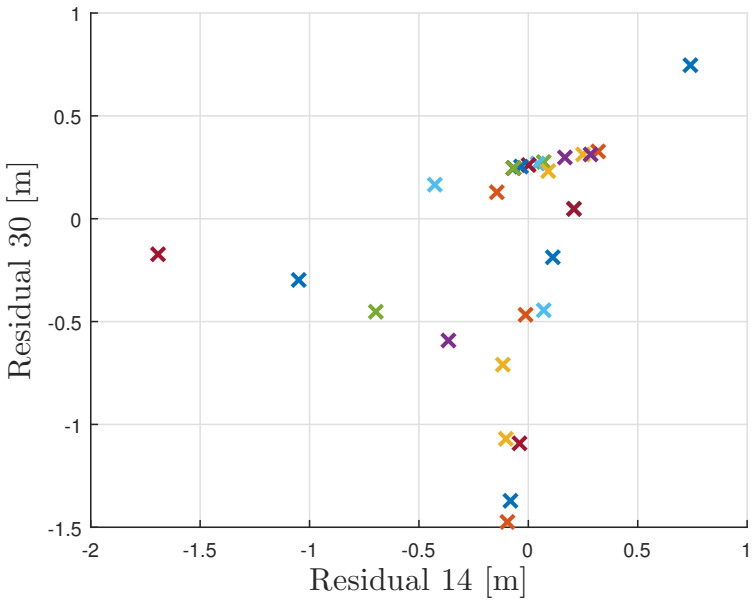
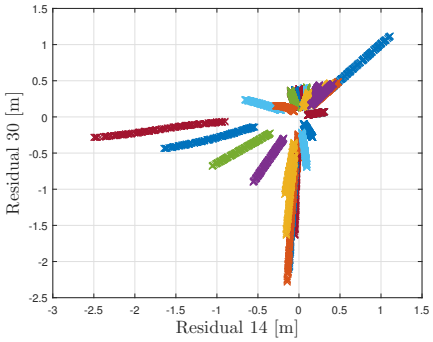


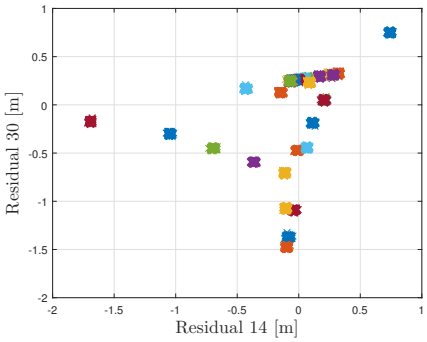
Figure 5.5: Residual space without any uncertainties in Hanoi WDN (each color represents a leak in a different location).

The leak uncertainty generated by (2.12) is shown in Figure 5.6a which produces a displacement of the residuals from the origin as the leak grows. Notice that not all the residuals follow a linear evolution (the nodes where the leaks can change their flow sign presents a more non-linear behavior), allowing the directional analysis of residuals (Casillas et al., 2012). In Figure 5.6c, the demand uncertainty generated by (2.9) is presented and produces a cloud of points around the scenario without uncertainty which diameter depends on the level of uncertainty. It should be noted that the demand uncertainty is the uncertainty that produces more overlapping between different leaks. In Figure 5.6b, the noise in measurements generated by

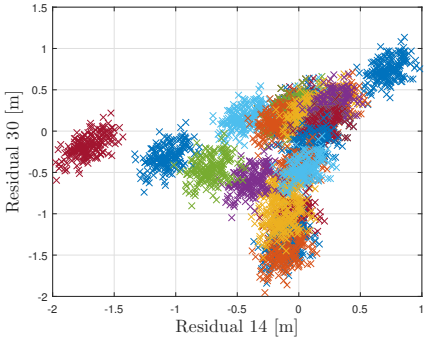
(2.7) is shown. It also needs to be noted that noise measurement produces the same effect as demand uncertainty but with less impact. Finally, in Figure 5.6d, the effect of all the uncertainties together is depicted.



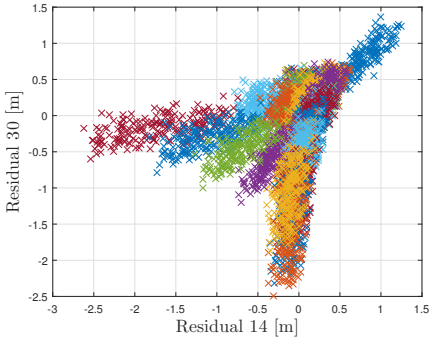
(a) Residual space with leak uncertainty in Hanoi WDN.



(b) Residual space with noise in Hanoi WDN.



(c) Residual space with demand uncertainty in Hanoi WDN.



(d) Residual space with all the uncertainties in Hanoi WDN.

Figure 5.6: Residual space with uncertainties and noise in Hanoi WDN (each color represents a leak in a different location).

Figure 5.7 shows the results of applying (2.13) to generate sequences of daily global water consumption demands.

Leak Localization

In order to illustrate the performance of the proposed methodology, four different studies have been carried out under the following particular conditions:

- A leak size uncertainty study considering a leak range between 25 and 75

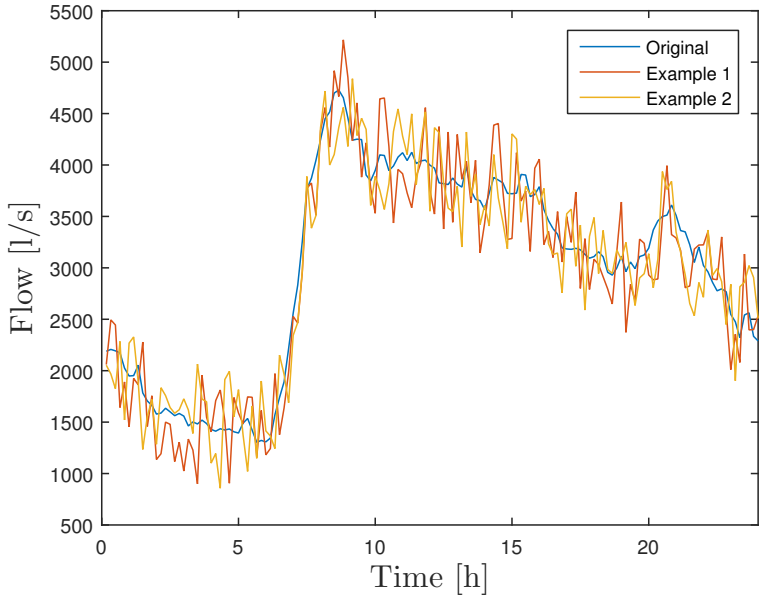


Figure 5.7: Original and artificially generated daily consumption.

l/s (0.84 and 2.51 % respectively, of the average of total amount of water demanded, which is 2991.1 [l/s]).

- A study considering noise in pressure measurements with an amplitude of ±5 % of the mean value for all pressure residuals.
- A demand uncertainty study considering an uncertainty of ±10 % of the nominal demand node values.
- A study considering that all the three uncertainties previously defined are simultaneously present in the WDN.

For each study, three complete data sets have been generated for each node (potential leak locations), one for training purposes, another used to generate the confusion matrix used in the k-NN time reasoning and the later one to test the leak localization performances. Each set used for testing, associated to a leak at a given node, is called a leak scenario. The variables conforming the data are the input flow $\tilde{d}_{WDN}$ and the two residuals r_{14} and r_{30} associated to the pressure measurements in nodes 14 and 30, respectively. The feature space used as input for the classifier is repre-

sented in Figure 5.8. The sampling time used in the simulations is 10 minutes, but hourly average values of variables are used to improve the leak location performance. Different daily input flow patterns have been simulated like the ones depicted in Figure 5.7. Accordingly to the scheme presented in Figure 5.1, the pressure residuals have been obtained by means of a WDN simulator (Epanet model of the network) where the uncertainties described above have been considered.

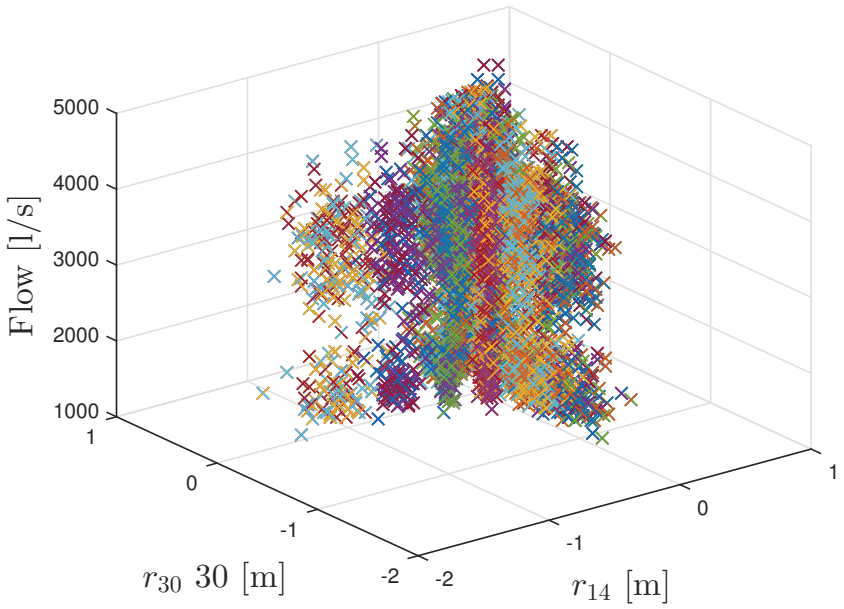


Figure 5.8: Residual space with uncertainties in Hanoi WDN (each color represents a leak in a different location).

In order to determine whether the three classifier inputs (r_{14}, r_{30} and $\tilde{d}_{WDN}$) follow a Gaussian distribution, a one-dimension Kolmogorov-Smirnov test on a training data set of 480 samples (for each of the 31 leak nodes) has been performed. As a result, the three inputs can be considered Gaussian distributed for a significance level of 3 %.

The results obtained by the proposed method in the four different studies have been compared to the ones obtained by using the leak-sensitivity analysis with the angle metric proposed by (Casillas et al., 2012) and summarized in Section 5.2. For the Angle method only the two residuals are used because the flow measurement has a

Table 5.1: Accuracy results in Hanoi WDN.

Study	$N = 1$			$N = 24$		
	k-NN	Bayes	Angle	k-NN	Bayes	Angle
Leak uncertainty	60.21	83.60	76.61	77.41	83.87	77.41
Noise in measurements	69.62	83.19	73.79	83.87	83.87	70.96
Demand uncertainty	31.18	39.11	41.39	58.06	45.16	64.51
All together	32.12	48.25	36.96	74.19	83.87	54.83

great value and tends to reduce the effect of residuals in the diagnosis, thus resulting in worse results. The sensitivity matrix (5.2) has been computed using (5.3) and by considering nominal leak conditions in every demand node ($\bar{v} = 0$, $\bar{d} = \hat{d}$ and $l_i = 50$ [l/s], $i = 1, ..., n_n$). The results obtained by using the Angle method and the two proposed methods, in both cases considering only one sample ($N = 1$) and the equivalent number of samples of one day ($N = 24$) in the leak localization diagnosis are summarized in Table 5.1. The values presented in this table correspond to the overall accuracy Ac defined in (2.16).

As it can be seen, the three methods provide good performance in the leak uncertainty case because of the linear directional variation of most of the residuals for this kind of uncertainty (Blesa et al., 2016). It must be noted that in the case that only demand uncertainty is considered, the classifier-based methods perform worse than when all the uncertainties are considered together. This happens because the leak uncertainty spreads the residual data providing a better separation (and for the Bayesian classifier the distribution tends to be more Gaussian).

When the time horizon and recursivity described in Subsection 5.6.1 are applied, it can be seen that there is an improvement in the performance achieved in all uncertainty cases (except for the case of the noise uncertainty for the Angle method, where the full performance is achieved since the first sample, and then fluctuates within the time horizon around the same values).

The effect of the horizon length N in the performance (Ac) for the three studied methods is also analyzed using the last study (to create the figures an extended data set, ten times larger, has been used). The results for the k-NN classifier are shown in Figure 5.9, while the results for the Bayesian classifier are shown in Figure 5.10,

and the results for the Angle method are shown in Figure 5.11. 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 predict the leak for which the diagnosis is still considered correct.

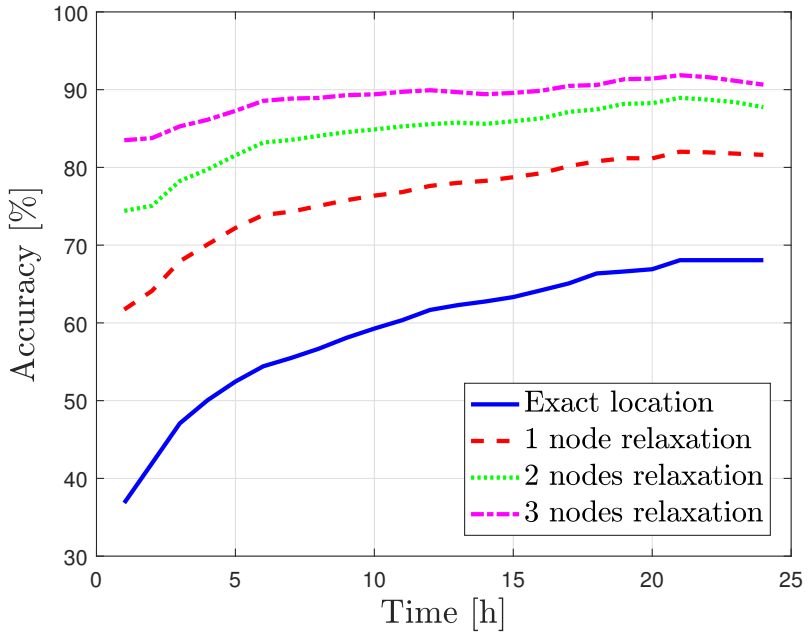


Figure 5.9: Accuracy results over a time horizon for the k-NN classifier in Hanoi WDN.

As expected, the accuracy increases with the time horizon length N. It can be observed that it reaches a steady state value when N is around twenty hours. This result justifies the use of a time horizon corresponding to one day and it agrees with the results already presented in (Casillas et al., 2012).

Finally, Figure 5.12 shows a comparison of the three studied methods by using a different performance indicator, the Average topological distance, which is the minimum distance in nodes between the node candidate and the node where the leak exists, as defined in Section 2.4.

The results show the good performance of both classifiers, especially the Bayesian classifier, which works better than the k-NN classifier when the data has a clear (in this case Gaussian) distribution (if not, the k-NN performs better as it can be seen

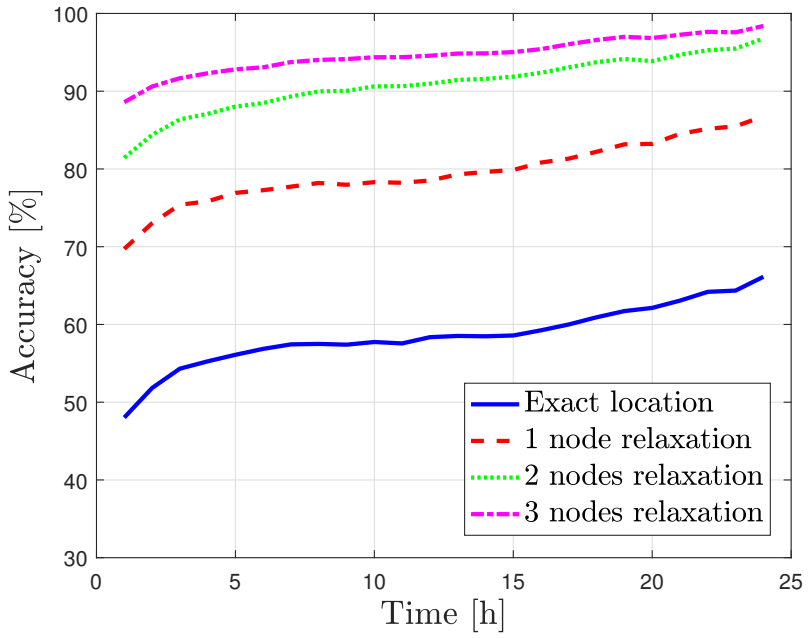


Figure 5.10: Accuracy results over a time horizon for the Bayesian classifier in Hanoi WDN.

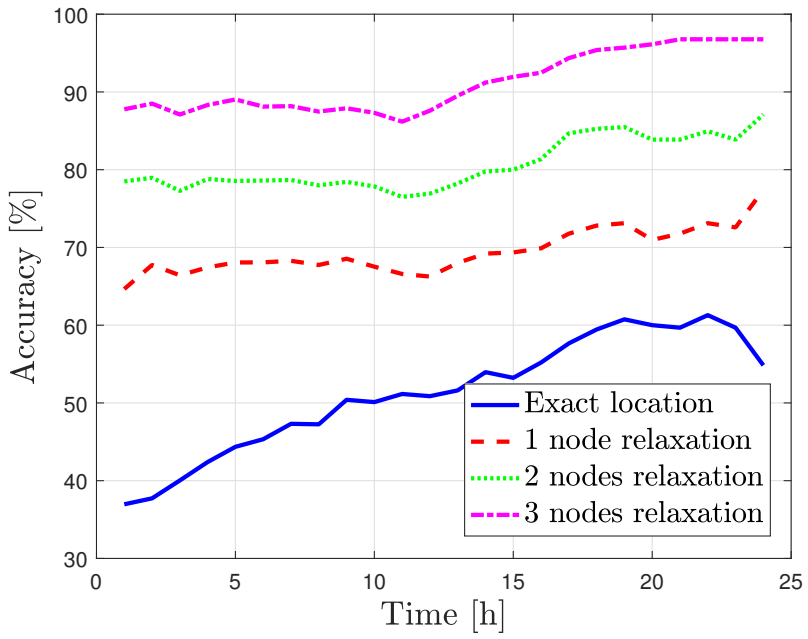


Figure 5.11: Accuracy results over a time horizon for the Angle method in Hanoi WDN.

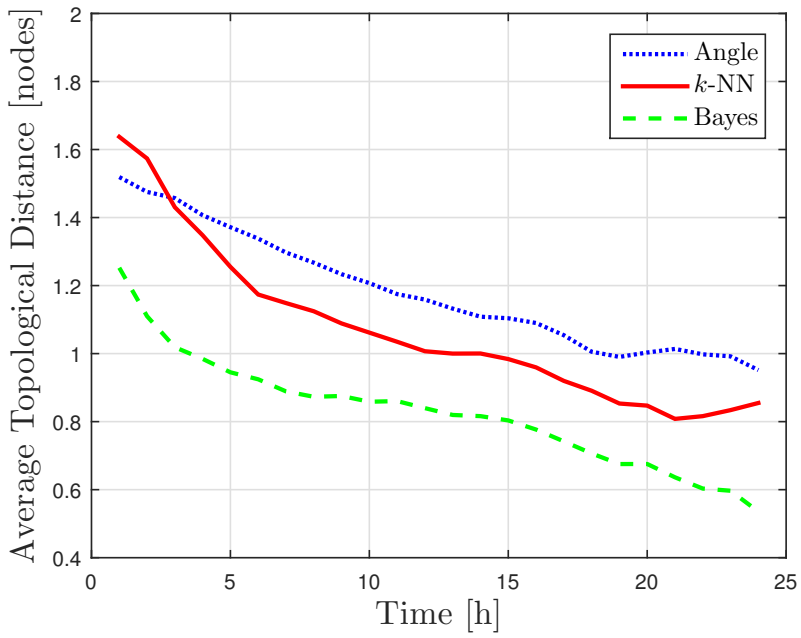


Figure 5.12: Average topological distance results in a time horizon in Hanoi WDN.

in the Table 5.1 for the demand uncertainty case), and also has a better reasoning over time. Also, in Figure 5.12, it can be seen that the Bayesian classifier tends to point a closer class when it fails than the k-NN classifier, but it can increase its performance at that point by choosing a bigger k value, but with a degradation of the exact localization (i.e., Ac) performance. To sum up, the Bayesian classifier should be used when the classes present a Gaussian distribution, and the k-NN classifier otherwise.

5.7.2. Nova Icària DMA Case Study

The two leak localization techniques proposed are tested in the Nova Icària DMA considering the real leak scenario described in Chapter 3. As in the previous case study, for all the measured variables, the average value of the six samples available each hour is used for leak localization purposes. Single leak scenarios have been considered in the 1520 nodes.

Here, the uncertainty modelling with real data explained in Section 2.1 is applied.

First, the system has been simulated considering the operating conditions of the leak-free scenario (input flow, boundary conditions and demand distributions). The differences between the 120 hourly samples of the five inner pressure sensors and the pressures estimated by the hydraulic model have been used to estimate the real uncertainty of the network (demand uncertainty, modeling errors and noise in the measurements). On the other hand, nominal hourly leak residuals $r_i^{(0)}(t)$ for $i = 1, \ldots, n_n$ and $t = 1, \ldots, 24$ have been computed as the difference of the estimated pressures in the five inner sensors in a leak scenario and the ones estimated in the normal operation.

A k-NN classifier (with $k = 3$) has been trained for leak localization and validated. The inputs of the classifier are: the five pressure residuals, the flow that enters into the DMA and the two set points of the valves. The data used in the training and testing stages are the 24 samples of nominal hourly residuals directly and adding the real uncertainty (120 samples): 96 samples for training and 48 for validation (for the generation of the confusion matrix $\mathbf{\Gamma}$). The same training data sets generated are used to calibrate the PDFs (assuming Gaussian distributions) for the Bayesian classifier.

Figure 5.13 shows the result of the two proposed methods after applying 24 hourly samples ($N = 24$): the k-NN classifier indicates that the leak is in node 3 while the real leak is in node 996, which means that the minimal topological distance is 13 nodes, and the geographical linear distance is around 184 meters. For the Bayesian classifier, the node candidate is 403 which has a minimal topological distance of 10 nodes and a geographical linear distance of 183 meters. As a comparison, the application of correlation method (Pérez et al., 2014) provides as node candidate the node 1036 (this result is also depicted in Figure 5.13), which is at a minimal distance of 17 nodes and 222 meters of the real leak location.

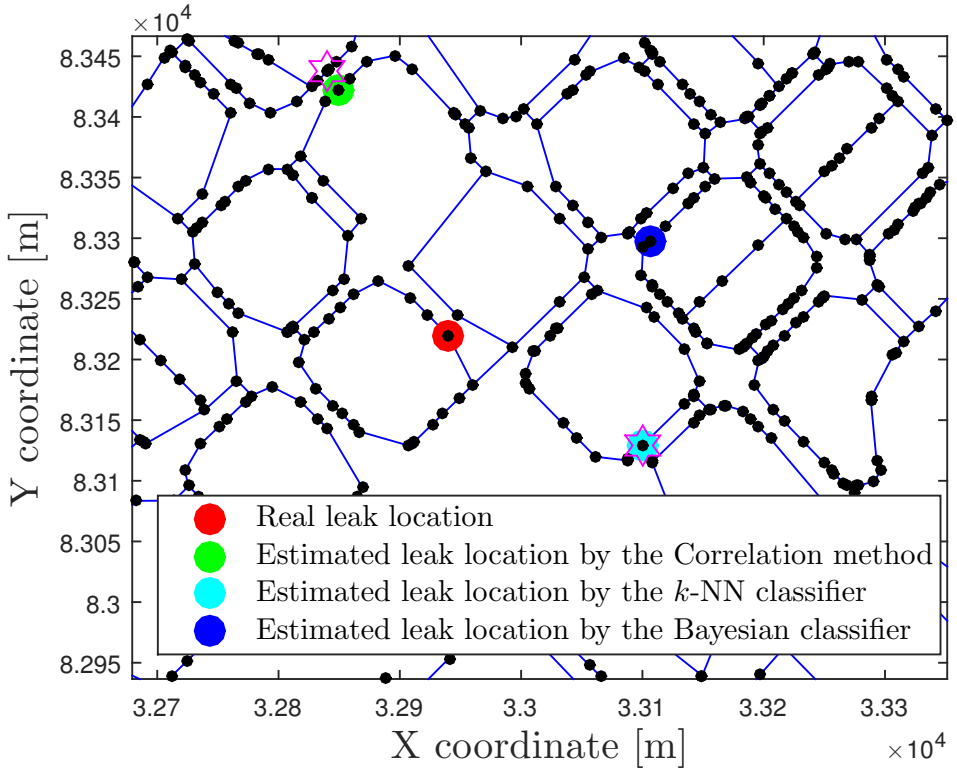


Figure 5.13: Comparison of different leak localization methods in Nova Icària DMA.

6. Data-Driven Leak Localization

In this chapter, a new data-driven method for leak localization in WDNs is proposed. The supervised regression technique based on the Kriging interpolation presented in Section 2.4 is used to generate estimated data from the sensed nodes and the topological information of the network. From these estimations, it is searched for the place the with largest evidence of leak that is proposed as the node candidate. Therefore, the use of the Bayesian temporal reasoning used in the model-based leak localization for the multi-class Bayesian classifier is used here to extract a unified and enhanced diagnosis from an ensemble of them.
This method is based in three principles:

- The use of historical measurements from pressure sensors placed across the network that are able to represent the behavior in nominal conditions of the network (i.e., measurements without leaks inside the network).
- The use of spatial interpolation (or regression) technique, in this PhD thesis the Kriging interpolation to estimate the pressure in the nodes where there are not sensors placed using the measurements from the sensed nodes.
- The use of the estimated pressure at each node of the network, where comparing the actual pressure measurements with leak with the historical pressure measurements without leak in similar operational conditions. The node with a largest fall of pressure by the leak is pinpointed as the node candidate.

From that, two main characteristics must be remarked. On the one hand, the leak localization approach is purely a data-driven approach that does not need the use

of a hydraulic simulator and the building and calibration of a network model. On the other hand, the proposed method only requires data in nominal operation, in contrast with other methods that require the data from the faulty conditions. These two characteristics make the proposed approach a fast and straightforward to apply in practice.

6.1. Assumptions and Basic Operation

As stated in Chapter 4 and Chapter 5, the leak detection and localization problem posed as a FDI problem in the literature assumes that only one leak can occur at a time. Additionally, it is also usually assumed that leaks can only occur in the nodes of the network (e.g., assumed in (Pérez et al., 2011) or (Casillas et al., 2012)), which makes the number of potential places that leaks can occur equal to the number of nodes of the network. Here, as in the two previous chapters, the same assumption is made.

We define as operating conditions $\mathbf{c}_d$ the positions of internal valves, reservoirs pressures, global consumption flow and users nodal demands. Note that here, in contrast to the boundary conditions $\mathbf{c}$ used in Chapter 5, the variable operational conditions $\mathbf{c}_d$ besides containing the boundary conditions $\mathbf{c}_d$ it also includes the nodal demands $\mathbf{c}_d$. Consider also the presence of a leak l_j with magnitude l and acting at node j. If pressure measurements are available in all the nodes of the WDN and historical data of these sensors is available for same operating conditions but in leak-free operation, then a residual vector $\mathbf{r}$ can be computed as

$$\mathbf{r} = \mathbf{p}(\mathbf{c}_d) - \mathbf{p}^{(l_j)}(\mathbf{c}_d) \tag{6.1}$$

where $\mathbf{r}$ is the vector of residuals in [m], $\mathbf{p}(\mathbf{c}_d)$ is the vector of pressure measurements in [m] in nodes of the WDN under operating conditions $\mathbf{c}_d$ in an scenario free of leaks while $\mathbf{p}^{(l_j)}(\mathbf{c}_d)$ is the vector pf pressure measurements in nodes of the WDN under the same operational conditions $\mathbf{c}_d$ but with a leak in node j with a leak size of l.

Using the residual vector $\mathbf{r}$, the leak localization can be performed by estimating the node where the leak is as the node with the highest residual component (see (Jensen and Kallesøe, 2016) and (Romano et al., 2017)) as

$$\hat{j} = \underset{j \in \{1,\ldots,n_n\}}{\arg\max} \{r_j\} \tag{6.2}$$

where r_j for $j = 1, \ldots, n_n$ are the components of residual vector $\mathbf{r}$ defined in (6.1). In practice, two limitations in the computation of (6.1) appear. The first limitation is that not all the nodes of the WDN have pressure sensors installed. Indeed, due to budget constraint only a few sensors are installed in consumer nodes inside the WDN. The current sensor placement in a WDN can be described by a vector $\mathbf{q}$

$$\mathbf{q} = (q_1, \ldots, q_{n_n}) \tag{6.3}$$

where n_n is the number of potential locations where sensors can be installed (i.e., all the nodes of the network) and components q_i are binary values that indicate if a sensor is placed at node i ($q_i = 1$) or not ($q_i = 0$). We define n_s as the number of sensors installed in the network. This limitation can be faced by using spatial interpolation techniques that starting from the available pressure measurements in some consumer nodes are able to estimate the pressure in the other consumer nodes without sensor.

The second limitation is due to the fact that the historical data is limited. So in practice, when a new measurement under the operational conditions $\mathbf{c}_d$ historical measurements with exactly the same operational conditions are not available. To overcome that problem, the use of a historical measurements with a closer operational conditions $\hat{\mathbf{c}}_d$ is considered instead. In practice, the flow and the pressure at the inlet are measured, but the nodal demands, unless AMRs are placed, are not measured.

Taking into account the two previous limitations, the ideal residual defined in (6.1)

can be approximated by

$$\hat{\mathbf{r}} = \hat{\mathbf{p}}(\hat{\mathbf{c}}_d, \mathbf{q}) - \hat{\mathbf{p}}^{(l_j)}(\mathbf{c}_d, \mathbf{q}) \tag{6.4}$$

where $\hat{\mathbf{p}}(\hat{\mathbf{c}_d}, \mathbf{q})$ is the vector that approximates the pressure map in the WDN under boundary conditions $\mathbf{c}_d$ and no leak scenario. The residual is computed by using pressure values $p_i(\hat{\mathbf{c}}_d)$ if $q_i = 1$ or otherwise by interpolation techniques to predict the pressure values for the unmeasured nodes. $\hat{\mathbf{p}}^{(l_j)}(\mathbf{c}_d, \mathbf{q})$ is the vector that approximates the pressure map in the WDN under operational conditions $\mathbf{c}_d$ and leak scenario of magnitude l in node j. This pressure is computed using actual measured values $p_i^{(l_j)}(\mathbf{c}_d)$ if $q_i = 1$ otherwise by interpolation techniques to predict the pressure values for the unmeasured nodes.

Then, in practice the approximated residual (6.4) will be used instead of the ideal residual (6.1) to perform the leak localization task as (6.2)

$$\hat{j} = \underset{j \in \{1,\ldots,n_n\}}{\arg\max} \{\hat{r}_j\} \tag{6.5}$$

It should be noted that the performance of the proposed approach depends on the number and location of the pressure sensors as long as the amount of historical data and their similarity with the current operational boundaries.

6.1.1. Pressure Estimation by Kriging Interpolation

To evaluate the residual (6.4), actual measured values available from the installed sensors and interpolation techniques to predict the pressure values for the unmeasured nodes will be used.

Here, the interpolation technique that is proposed to be used is based on the use of the Kriging interpolation approach, which is broadly used in the field of geostatistics (Kleijnen, 2017), and has been described in Section 2.4. The basic idea of Kriging interpolation is to predict the value of a function at a given point by computing a

weighted average of the known values of the function in the neighborhood of the point. The method can be seen as a multivariate regression approach.
The estimation of an unmeasured pressure in a node i is given by a fitted Kriging interpolation model with parameters μ and $\varepsilon(\cdot)$ is

$$\hat{p}_i^{(l_j)}(\mathbf{c}_d, \mathbf{q}) = \mathfrak{H} + \varepsilon(\mathbf{p}^{(l_j)}(\mathbf{c}_d, \mathbf{q}), \mathfrak{d}_i(\mathbf{q})) \tag{6.6}$$

where $\mathfrak{H}$ is a constant that represents the constant part of the interpolation and function $\varepsilon(\boldsymbol{\chi}, \boldsymbol{\theta}, \mathfrak{d}_i(\mathbf{q}))$ is the spatially correlated part of the variation. Both terms, constant $\mathfrak{H}$ and function $\varepsilon(\cdot)$ are obtained in the fitting process as well as function parameters $\boldsymbol{\chi}$ and $\boldsymbol{\theta}$. On the other hand, $\mathfrak{d}_i$ is the i^{th} row of a symmetric matrix $\mathfrak{D} \in \Re^{n_n \times n_n}$ whose components $d_{i,j}$ are the minimum distance in pipe [m] from node i to node j and $(\mathbf{q})$ denotes that only the components of $\mathfrak{d}_i$ associated to the measured nodes are considered. The fitting process consist in a least square error minimization problem considering available pressure measurements $\mathbf{p}^{(l_j)}(\mathbf{c}_d, \mathbf{q})$ and distance matrix $\mathfrak{D}_s(\mathbf{q}) \in \Re^{n_s \times n_s}$ which is a submatrix of the matrix $\mathfrak{D}$ but only considering the distances between the n_s sensors.

6.1.2. Bayesian Time Reasoning

With the aim of improving the performance of the leak localization approach defined in (6.5), the residual vector (6.4) is updated at each time instant t for every leak node candidate $i = 1, \ldots, n_n$. First, and in order to have only positive values, the smallest residual value is added to all the residuals as

$$\hat{r}_i^{(+)}(t) = \hat{r}_i(t) - \min(\hat{\mathbf{r}}(t)) \tag{6.7}$$

The $\mathfrak{S}(t)$ likelihood index is then obtained for every leak node candidate $i = 1, \ldots, n_n$ as the normalization of the $\hat{r}_i^{(+)}(t)$ values as follows

$$\mathfrak{S}_i(t) = \frac{\hat{r}_i^{(+)}(t)}{\sum_{j=1}^{n_n} \hat{r}_j^{(+)}(k)} \tag{6.8}$$

The value of $\mathfrak{S}_i(t)$ is a measure of which ones of the n_n nodes present the most disturbed residuals and therefore are candidates to be the leaking node. This information can be combined to the initial leak probabilities for each node $P_i(t-1)$ by means of the Bayes rule in order to obtain updated posterior leak probabilities $P_i(t)$

$$P_i(t) = \frac{P_i(t-1)\mathfrak{S}_i(t)}{\sum_{j=1}^{n_n} P_j(t-1)\mathfrak{S}_j(t)} \tag{6.9}$$

Then, the leak node localization can be estimated by using posterior leak probabilities instead the evaluation of approximated residual vector in (6.5) by

$$\hat{j}(t) = \underset{i\in\{1,\ldots,n_n\}}{\arg\max} \{P_i(t)\} \tag{6.10}$$

The starting point can be an unprejudiced one (if no further information is available), i.e., $P_i(t) = 1/n_n$, and as long as new sample measurements are available and the $\mathfrak{S}_i(t)$ are computed, the updated values of $P_i(t)$ allow to go on discarding many of the competing leaks.

One drawback is that if any of the leaks takes the posterior probability value of 1 at any t, then all the remaining leaks take the 0 probability value, therefore preventing them to have a future value different from zero due to the recursive application of (6.9). Similarly, if a leak node takes a zero value at any time instant that will be the result, which given the normalization process, this is the case for one node at each time instant. To avoid this problem, a term corresponding to the average value multiplied by the parameter $\mathfrak{Z}$ is added in the normalization process so that (6.8) is improved as

$$\mathfrak{S}_i(t) = \frac{\hat{r}_i^{(+)}(t) + \frac{\mathfrak{Z}}{n_n}\sum_{z=1}^{n_n}\hat{r}_z^{(+)}(t)}{\sum_{j=1}^{n_n}(\hat{r}_j^{(+)}(t) + \frac{\mathfrak{Z}}{n_n}\sum_{z=1}^{n_n}\hat{r}_z^{(+)}(t))} \tag{6.11}$$

This way the maximum value of $\mathfrak{S}_i(t)$ at any time instant t is limited to

$$\mathfrak{S}_{max}(t) = 1 - \frac{\mathfrak{Z}\frac{n_n-1}{n_n}\sum_{z=1}^{n_n}\hat{r}_z^{(+)}(t)}{\sum_{j=1}^{n_n}(\hat{r}_j^{(+)}(t) + \frac{\mathfrak{Z}}{n_n}\sum_{z=1}^{n_n}\hat{r}_z^{(+)}(t))} \tag{6.12}$$

and the minimum value for $\mathfrak{S}_i(t)$ at any time instant t is guaranteed to be at least

$$\mathfrak{S}_{min}(t) = \frac{\frac{3}{n_n}\sum_{z=1}^{n_n}\hat{r}_z^{(+)}(t)}{\sum_{j=1}^{n_n}(\hat{r}_j^{(+)}(t) + \frac{3}{n_n}\sum_{z=1}^{n_n}\hat{r}_z^{(+)}(t))} \tag{6.13}$$

6.1.3. Summary

The application of the methodology comprises both off-line and on-line stages. The off-line stage is limited to the application of a sensor location algorithm, as the one presented in the next Chapter 7 for determining the optimal location of a given number of pressure sensors to be installed in internal network nodes. At the online stage, at each time instant t, several steps are required. First, the current operating conditions and internal pressures are determined. If the leak detection module (the implementation of this module is out of the scope of this paper) determines that the network is not affected by any leak, then the current operating conditions and internal pressure values are used to actualize a database that stores an always up-to-date historical normal operation data set. On the other hand, if the fault detection module indicates the presence of a leak, then the leak localization procedure detailed in the previous subsections is triggered:

- Look for leak-free historical data captured under similar operating conditions, i.e., at the same hour of the day and with similar input pressure and flow.
- Apply Kriging spatial interpolation (6.6) to the selected historical values to obtain the reference pressure map, i.e., a map containing the reference pressure values for all the network nodes.
- Apply Kriging (6.6) to the measured values to obtain the current pressure map.
- Compare the current and the reference pressure maps computing the residual (6.4).

- Identify the leaky node as the one for which the difference between pressure maps is highest by using (6.5).
- Integrate the individual diagnosis in a time horizon scheme to improve the performance by means of the Bayes rule (6.9).

6.2. Case Studies

The proposed data-driven leak localization approach is tested in three networks. First, the Hanoi benchmark network is used to experimentally verify the assumption of (6.2) by means of artificial data. Then, two real cases are presented, the Nova Icària DMA in which the method is compared with the ones obtained in the Chapter 5 and a new real case from the Pavones DMA network.

6.2.1. Hanoi WDN Case Study

This network, presented in Chapter 3 is used to justify the validity of (6.2) and to test the performance using artificial data.

Leak Localization Proof of Concept

Using the Epanet hydraulic simulator data from all sensors with an artificial leak or without the leak under the same operational conditions $\mathbf{c}_d$ has been generated to compute the residuals as stated in Equation 6.1. Using this data the normalized sensitivity matrix (i.e., the columns are the residual output of the measurements of the node with that column index and the rows the node with leak) can be build for a leak of 100 [l/s] where it can be seen that the biggest pressure changes are at the node where the leak is generated for all leaks as depicted in Figure 6.1. The rows of the matrix are independently scaled to have their values from 0 to 1.

A particular case can be used to illustrate the generated pressure maps. In Figure 6.2, the reference pressure map, the current pressure map, the residual map, and also the residual profile, for a leak in node 16 are shown.

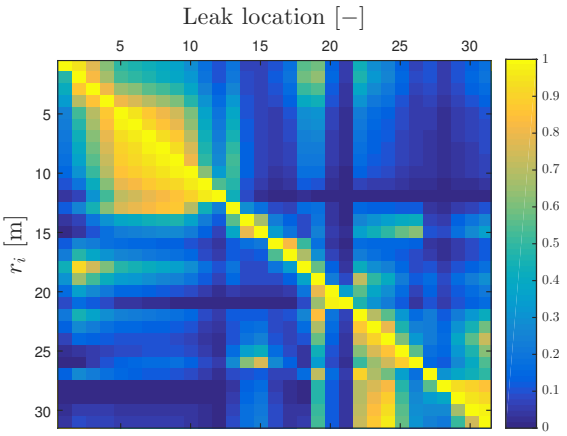
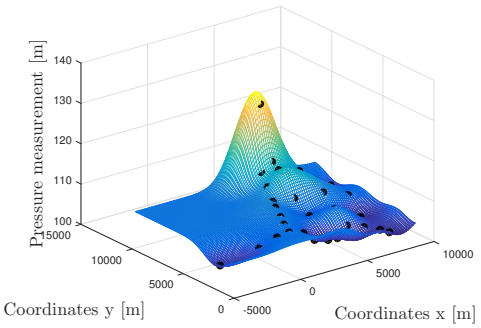
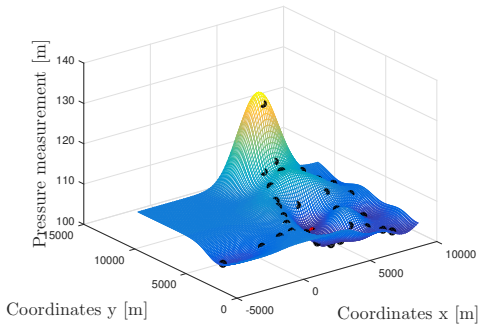


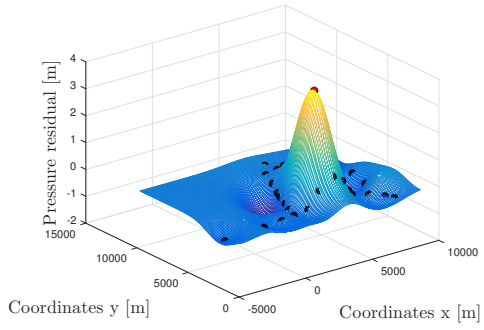
Figure 6.1: Normalized sensitivity matrix of the Hanoi WDN for a leak of 100 [l/s] with all the sensors.



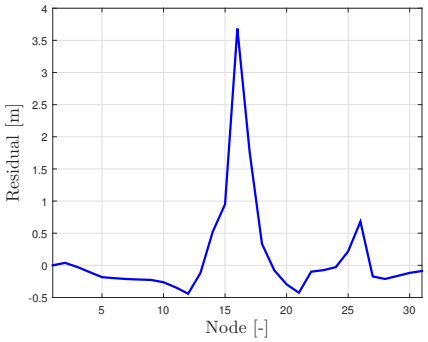
(a) Pressure map without leak.



(b) Pressure map with leak.



(c) Residual pressure map.



(d) Residual pressure profile.

Figure 6.2: Kriging interpolation for the case of a leak of 100 [l/s] at node 16.

Table 6.1: Leak localization results in the simplified Hanoi WDN.

Time horizon	ATD
N=1	3.2
N=24	2.2

Leak Localization

Using the hydraulic simulator, data with the following uncertainties are created:

- The leak uncertainty is considered by not knowing the exact leak size, but knowing that it is contained in the range of 25 and 75 [l/s].
- The noise in the measurements is considered, where a white noise of the amplitude of 0.1 (zero mean) meter water column ([mwc]) is added.
- The demand uncertainty is considered, by introducing an uncertainty of the 10 [%] of the nominal demand value.

The daily global consumption pattern is generated as described in Chapter 5 for six sensors. The sampling rate is 10 minutes, but the measurements at each hour are averaged to reduce the impact of the uncertainties in the diagnosis stage.
The sensor placement used here is the result of the proposed incremental feature selection technique presented in Chapter 7. The application of the proposed leak localization method to the testing data is summarized in the Table 6.1 for the time horizons $N = 1$ (one hour) $N = 24$ (one day), where "ATD" is the average topological distance (i.e., the average error of the leak localization performance) in [nodes]. The complete ATD diagnosis performance for $N = 1$ to $N = 24$ is shown in Figure 6.3.
It must be remarked that in the interpolation of the estimated pressures $\hat{\mathbf{p}}$ made by the Kriging, the coordinates are not used to map the pressures, instead the minimum distance in pipe is used in order to avoid closer nodes in coordinates, but connected through a lot of meters of pipes.
Also to identify the reference map, past measurements took at the same hour are used to try to catch the daily routine consumption of water to try to minimize the

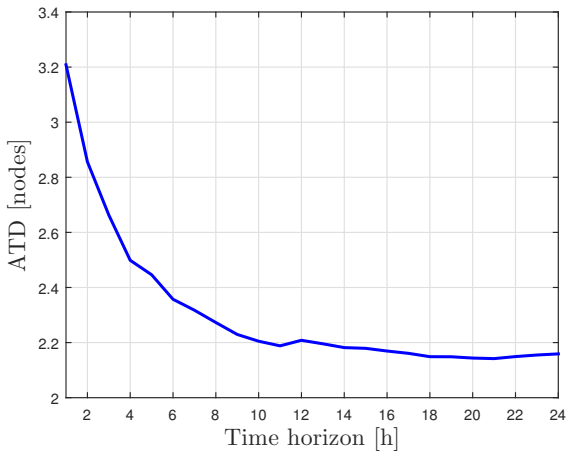


Figure 6.3: ATD results of the proposed leak localization approach in the Hanoi WDN.

nodal demands differences, which are unknown. And from these measurements, the ones with closest global consumption with respect to the actual measurements are used.

6.2.2. Nova Icària DMA Case Study

The Nova Icària DMA network presented in Chapter 3 and already used in Chapter 5 to test the k-NN classifier and the Bayesian classifier is used to assess the data-driven technique proposed in this chapter.

Here, the real case is used considering that the first five days of data (hourly averaged) without leak are used as the historical data to create the interpolated maps to be compared with the maps generated with the actual pressure measurements. All the node candidates obtained for the different methods and the location of the real leak are presented in the Figure 5.13.

The result of the diagnosis using one day of data with leak (the first 24 hours) provides as the node candidate the one with index 7, which is in a geometric distance of 239.9 meters and a topological distance of 23 nodes from the leaky node. This result is worst but close to the ones obtained using model-based methods, which are, the correlation method presented in (Pérez et al., 2014) that provides as a node

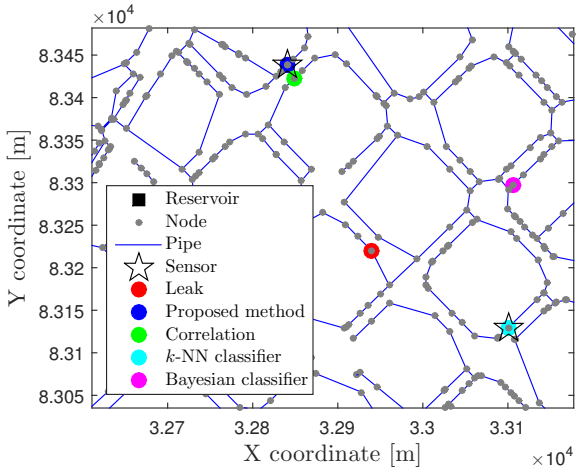


Figure 6.4: Nova Icària DMA leak localization results.

candidate the node 1036 with a linear geometric distance of 222.0 meters and a topological distance of 17 nodes. The k-NN method presented has as node candidate the node 3, with a linear geometric distance of 184.0 meters and a topological distance of 13 nodes. Finally, the most recent method tested in this real case, that also use Bayesian reasoning has as node candidate the node 403 with a linear geometric distance of 183.2 meters and a topological distance of 10 nodes.

6.2.3. Pavones DMA Case Study

The Pavones DMA network and its real case are described in Chapter 3. As proposed earlier, the data is reduced to be hourly by averaging all the samples in each hour. The interpolation is done using the same data model adjust as the two previous examples. The data used for the training data set corresponds to the measurements taken from the day 25^{th} of November of 2016 at 03:00 pm to the 29^{th} of November of 2016 at 00:58 am. The data used to do the localization of the leak starts at the 29^{th} of November of 2016 at 04:00 am and finishes the 1^{st} of December of 2016 at 09:58 am (54 hours).

The real leak location and the node candidate resulting of the leak localization method proposed are depicted in Figure 6.5, where the geometric distance between

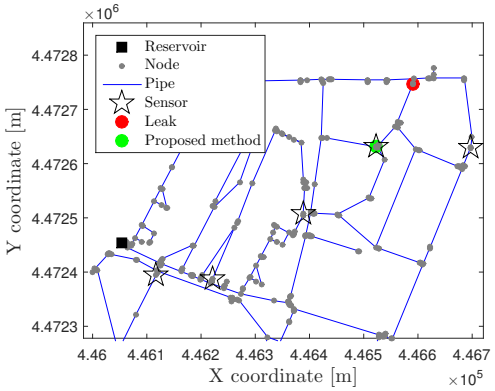


Figure 6.5: Pavones DMA leak localization results.

the two nodes is 134 meters and the topological distance is 8 nodes.

7. Sensor Placement

Given the limited number of sensors that can be deployed inside the WDNs due to budget constraints, their location inside the WDN is critical to achieve the best performance possible, which usually is particular for every leak localization method. In this chapter, the sensor placement approaches for the leak localization methods presented in Chapter 5 and Chapter 6 are proposed with the aim of maximizing their performance.

Three sensor placement methods based on Feature Selection (FS) techniques are proposed in this chapter. First, a more traditional approach intented to be used with the leak localization based on classifiers presented in Chapter 5 where a wrapper is applied with the aim of maximizing the classification accuracy (Ac). Then, this approach is modified and enhanced with the use of metrics particular of leak localization problem (ATD indicator) where a hybrid feature selection is presented where the combination of a filter in a first stage, the FCBF presented in Section 2.4 is used, and a wrapper, the GA also presented on Section 2.4, in the second stage is applied. Finally, an incremental feature selection method, the SFFS approach presented in Section 2.4, more suitable in computational load terms when a large number of sensors are needed to be placed, as e.g., in the case of the data-driven leak localization approach presented in Chapter 6.

In this chapter, the problem of sensor placement is formulated as a FS problem (Tang et al., 2014). Feature (or variable/attribute) selection techniques (Guyon and Elisseeff, 2003) are used to identify a subset of relevant variables in a data set, regarding its use to build a model with a given purpose, for instance a classifier.

Within the framework proposed in Chapter 5, the main idea is to generate, using a hydraulic simulation of the considered WDN, a complete data set containing all the potential residuals associated to the network nodes and apply a FS algorithm that determines the ones that after training the classifier will provide the best leak localization results.

In addition to the n_s selected features (pressure residuals of inner nodes), as was described in previous chapters and as can be seen in Figure 5.1, the classifier is also fed with the measured boundary conditions of the network $\tilde{\mathbf{c}}$ that will provide n_b fixed additional 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.

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 one of the leak localization method schemes presented in the previous chapters. 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 select the best subset of features.

To select a configuration with n_s sensors, the following binary vector, already introduced in Chapter 6 is defined

$$\mathbf{q} = \left(q_1, \ldots, q_{n_f}\right) \tag{7.1}$$

where $q_i = 1$ if the pressure in the node i is measured, and $q_i = 0$ otherwise (i.e., the vector $\mathbf{q}$ denotes which sensors are installed). n_f is the number of features, in this case the potential locations to place sensors, which for pressure sensor is the consumer nodes. So, in this work is assumed that $n_f = n_n$, but in practice usually not all the nodes are suitable places to install the sensors. Thus, this number can be reduced to a subset of these possible places determined by the WDN management company.

7.1. Optimal Sensor Placement for Classifiers

In order to evaluate the quality of a sensor configuration regarding its ability to locate a leak at node $i \in \{1, \ldots, n_n\}$, and assuming the case of a single leak, the first performance index to optimize is the classifier accuracy described in (2.16).
This performance index depends on the configuration of sensors considered that is parameterized in terms of the binary variable $\mathbf{q}$ to determine the best selection, which is

$$\mathrm{Ac}(\mathbf{q}) = \frac{\sum_{i=1}^{n_n} \Gamma_{i,i}(\mathbf{q})}{\sum_{i=1}^{n_n} \sum_{j=1}^{n_n} \Gamma_{i,j}(\mathbf{q})} \tag{7.2}$$

Note that for a given sensor configuration $\mathbf{q}$, $100\mathrm{Ac}(\mathbf{q})$ is the percentage of correctly localized leaks.
Based on the vector $\mathbf{q}$ and the performance index $\mathrm{Ac}(\mathbf{q})$, the sensor placement problem can be translated into an optimization problem formulated as follows

$$\max_{\mathbf{q}} \mathrm{Ac}(\mathbf{q}) \tag{7.3}$$

s.t.

$$\sum_{i=1}^{n_f} q_i = n_s$$

where $q_i \in \{0, 1\}$ is defined in (7.1) and $n_s \in \{1, \ldots, n_n\}$ is the number of sensors that we want to place.

7.1.1. 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

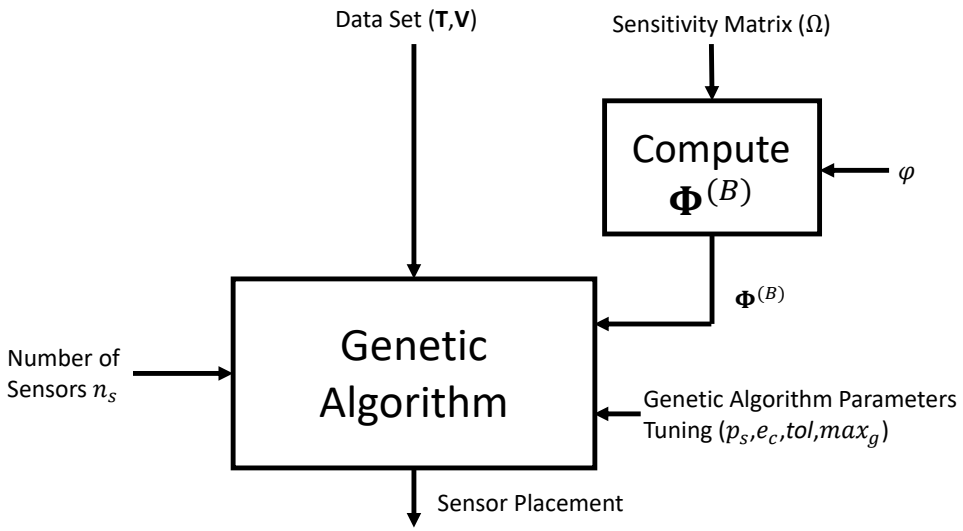


Figure 7.1: Scheme of optimization process.

implemented using the Global Optimization Toolbox of MATLAB.

The overall procedure can be seen in the Figure 7.1, where the "*Nominal Sensitivity Matrix*" is a data set containing only residuals without uncertainties.

7.1.2. Sensor Distance Matrix

In order to reduce the amount of sensor configurations to be tested in the GA heuristic search, sensor configurations (defined by $\mathbf{q}$) 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 Chapter 5 in (5.3) 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 $\hat{\mathbf{d}}$ is used) and nominal leak size ($l^{(0)}$) (Blesa et al., 2014). A difference of the sensitivity matrix presented in (5.3), here the sensor matrix is complete (i.e., sensors in all nodes are placed).

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 (Sarrate et al.,

2014a). If we consider a nominal approximated sensitivity matrix

$$\mathbf{\Omega} = \begin{pmatrix} \mathfrak{s}_1 \\ \vdots \\ \mathfrak{s}_{n_n} \end{pmatrix} \tag{7.4}$$

where $\mathfrak{s}_i$ for $i = 1, ..., n_n$ are row vectors

$$\mathfrak{s}_i = (\Omega_{i,1}, \ldots, \Omega_{i,n_n}) \tag{7.5}$$

with components computed using (5.3), a sensor distance matrix $\mathbf{\Phi}$ can be defined as

$$\Phi_{i,j} = \|\mathfrak{s}_i - \mathfrak{s}_j\|_1 \text{ for } i = 1, \ldots, n_f \text{ and } j = 1, \ldots, n_f \tag{7.6}$$

$\mathbf{\Phi}$ is a symmetric square matrix of dimension n_f and diagonal 0. A threshold φ can be determined in order to decide whether two sensors have a different behavior in the residual space or not. Then, a binary matrix $\mathbf{\Phi}^{(B)}$ 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

$$\Phi_{i,j}^{(B)} = \begin{cases} 0 & \text{if } |\Phi_{i,j}| < \varphi \\ 1 & \text{if } |\Phi_{i,j}| \geq \varphi \end{cases} \tag{7.7}$$

7.1.3. Data Format

The algorithms presented in the following subsections assume that the data generated after the simulation of the network under different conditions are organized in a particular way. Hence, both the training $\mathbf{T}$ and validation $\mathbf{V}$ data matrices are three-dimensional matrices, as shown in Figure 7.2. The first dimension of those matrices is associated to the features, i.e., to the n_b boundary measurements and the n_f pressure residuals computed at the different nodes of the network where pressure sensors can be installed. The second dimension is associated to the classes (with

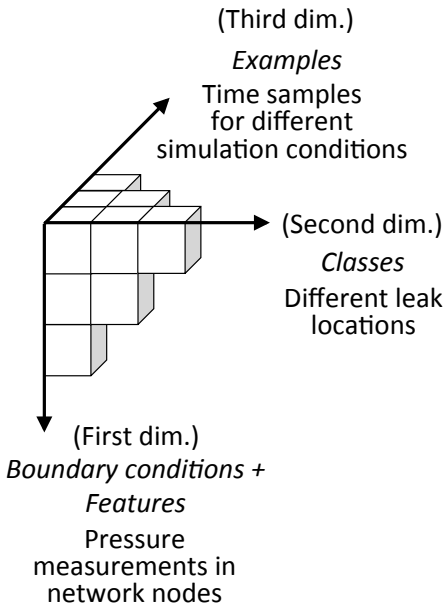


Figure 7.2: Data format.

size n_c), i.e., to the different possible leak locations that are considered. Finally, the third dimension is associated to the examples: for a fixed residual and a fixed leak location, the third dimension collects residual values that are obtained under different leak sizes, node demand estimations, noise realizations and at different time instants. The length of this last dimension will be denoted as m_T and m_V for the training and validation matrices.

Sensor Placement Algorithm

The optimization problem (7.3) solved by the genetic algorithms has as objective function to be optimized the accuracy defined in (2.16). The accuracy of the current evaluated configuration of sensors will be assessed after the classifier training process has ended by using a validation data set obtained from the hydraulic simulator as described in Section 2.1. A training matrix $\mathbf{T}$ and a validation matrix $\mathbf{V}$ 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.

Two modifications have been included into the basic GA scheme to increase its speed. On the one hand, the information of the objective function is stored in the case that the members that appeared earlier appear again. In such a case the stored value is retrieved instead of calculating the fitness function again (as proposed in (Oh et al., 2004)). On the other hand, the matrix $\mathbf{\Phi}^{(B)}$ is used to avoid fitness functions calculations. Hence, when a combination contains a not allowed pair of features, the worst value is directly assigned without the computation of the fitness function.

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 tol, the population size p_s, the elite count e_c in order to save part of the previous analyzed results and the maximum number of generations allowed max_g. 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 $\mathbf{I}$ is delivered by the GA (line 5) in the first generation. This matrix is obtained from the results of the past generation and the crossing and mutation parameters. Matrix $\mathbf{I}$ with the population of configuration of sensors is passed to the matrix $\mathbf{Q}$, then at each iteration in the current generation t one configuration of this matrix (row of the matrix), the vector $\mathbf{q}_t$, is evaluated. 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. 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 (7.7) 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

Algorithm 1 Sensor placement based on Genetic Algorithms.

Require: A training matrix $\mathbf{T}$ and a validation matrix $\mathbf{V}$. The number of features to select n_s, the number of nodes n_n, the population size p_s, the elite count e_c, the maximum number of generations allowed max_g, the tolerance to determine when the solution has reached *tol* and the binarized matrix $\mathbf{\Phi}^{(B)}$.

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

1: $init \leftarrow \mathrm{InitVarGA}(p_s, e_c, tol, max_g)$
2: $constraint \leftarrow \mathrm{SetConstraints}(n_f, n_s)$
3: **Inputs:** *init*, *constraint*, $\mathbf{T}$, $\mathbf{V}$, p_s, n_n, $\mathbf{\Phi}^{(B)}$.
4: **while** An optimization criterion is not reached **do**
5: GA based search:
6: Generate $\mathbf{I}$ matrix of size $(p_s \times n_f)$ where each row is a member of a generation.
7: $\mathbf{Q} \leftarrow \mathbf{I}$
8: **for** $t = 1, \ldots, p_s$ **do**
9: **if** $\mathrm{GetUsed}(\mathbf{q}_t) = 0$ **then**
10: **if** $\mathrm{CheckCombinations}(\mathbf{\Phi}^{(B)}, \mathbf{q}_k) = 1$ **then**
11: $\mathrm{Classifier}(\mathbf{q}_t) \leftarrow \mathrm{Train}(\mathbf{T}(\mathbf{q}_t))$
12: $\Gamma(\mathbf{q}_t) \leftarrow \mathrm{Validate}(\mathrm{Classifier}(\mathbf{q}_t), \mathbf{V}(\mathbf{q}_t))$
13: $\mathrm{Ac}(\mathbf{q}_t) \leftarrow \frac{\sum_{i=1}^{n_n} \Gamma_{i,i}(\mathbf{q}_t)}{\sum_{i=1}^{n_n} \sum_{j=1}^{n_n} \Gamma_{i,j}(\mathbf{q}_t)}$
14: $\mathrm{SetUsed}(\mathbf{q}_t)$
15: $\mathrm{SetAc}(\mathrm{Ac}(\mathbf{q}_t), \mathbf{q}_t)$
16: **else**
17: $\mathrm{Ac}(\mathbf{q}_t) = 0$
18: **end if**
19: **else**
20: $\mathrm{Ac}(\mathbf{q}_t) = \mathrm{GetAc}(\mathbf{q}_t)$
21: **end if**
22: **end for**
23: Find $\{\mathbf{q}, \mathrm{Ac}_{max}\}$ such that $\mathrm{Ac}_{max} = \max_{\mathbf{q}}(\mathrm{Ac}(\mathbf{q}_1), \ldots, \mathrm{Ac}(\mathbf{q}_{p_s}))$.
24: **end while**

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 $\mathbf{q}$ allows the selection of the adequate columns of the matrices $\mathbf{T}$ and $\mathbf{V}$ 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 $\mathbf{I}$, 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.

7.1.4. Hybrid Feature Selection Approach

Here, a second and more particular to the application approach of sensor placement in WDN is proposed for the classifier-based leak localization methods presented in Chapter 5. The proposed solution of the sensor placement problem aims at selecting a subset of relevant and non-redundant features (variables) for use in the classifier construction.

As it was previously discussed, we will consider that it is possible to place a sensor in all the nodes of the network, i.e., $n_f = n_n$.

In order to evaluate the quality of a feature selection regarding the leak localization performance, the average topological distance index (2.17) will be used here instead of the accuracy index described in (2.16) since it is a better index to assess the quality of the leak localization performance. This performance index depends on the configuration of features considered and it is parameterized in terms of the $\mathbf{q}$ vector (7.1) to determine the best selection

$$\mathrm{ATD}(\mathbf{q}) = \frac{\sum_{i=1}^{n_f}\sum_{j=1}^{n_f}\Gamma_{i,j}(\mathbf{q})\mathbf{D}_{i,j}}{\sum_{i=1}^{n_f}\sum_{j=1}^{n_f}\Gamma_{i,j}(\mathbf{q})} \tag{7.8}$$

Based on the vector $\mathbf{q}$ and the performance index $\mathrm{ATD}(\mathbf{q})$, the feature selection problem can be translated into an optimization problem formulated as follows

$$\begin{aligned} &\min_{\mathbf{q}} \mathrm{ATD}(\mathbf{q}) \\ &\text{s.t.} \\ &\sum_{i=1}^{n_f} \mathrm{q}_i = n_s \end{aligned} \tag{7.9}$$

where $\mathbf{q}$ is defined according to (7.1) and $n_s \in \{1, \ldots, n_f\}$ is the given number of sensors (features) to be selected.

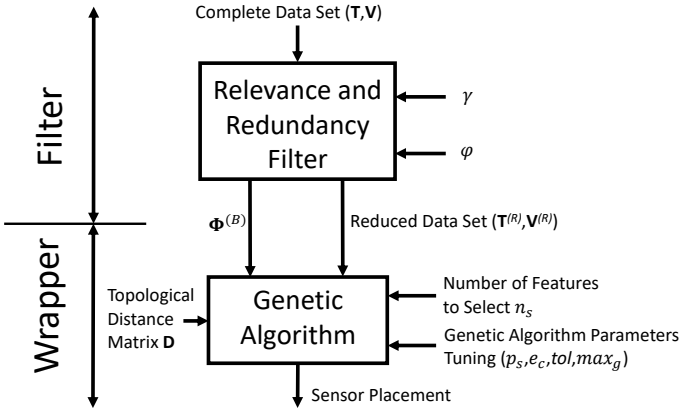


Figure 7.3: Hybrid feature selection scheme.

Overview

The proposed method is a hybrid approach with two different stages that are performed in a sequential way. First, an initial reduction of the dimension is performed by using a filter-based on evidences of the relevancy and redundancy of the variables that additionally assesses information about suitable/unsuitable pairs of combinations of features. Second, the subset of features that remains after the filter and the additional information is taken into account by the proposed wrapper method, which is a genetic algorithm, to tackle the combinatory problem and obtain a suboptimal feature selection. The whole procedure is depicted in Figure 7.3.

Filtering

With the aim of reducing the computational load of the wrapper, an initial reduction of the n_f dimension of the original feature space $\mathbf{F}$ of inner pressure residuals is applied. This dimensionality reduction is based on a relevance/redundancy-based filter that removes the most irrelevant and similar features. This reduction increases the performance of the proposed wrapper strategy which is based on the use of genetic algorithms. The proposed filter is based on the FCBF presented in (Yu and Liu, 2004), where first the features are ranked based on their relevance and then a sequential procedure to remove the redundant (and less relevant) features is

performed.

Relevance metric

Relevance is associated to the information that a given feature possesses according to the final problem to be solved. For classification problems, relevance is associated with the variability of the feature across the classes. Hence, in order to compute the relevance of each feature, the average training matrix $\mathbf{\Lambda}$ with size $n_f \times n_f$ (considering $n_f = n_c$) and associated to inner pressure residuals is defined with features as rows and classes as columns

$$\Lambda_{i,j} = \frac{1}{m_T} \sum_{z=1}^{m_T} \mathrm{T}_{(n_b+i),j,z} \text{ for } i = 1, \ldots, n_f \text{ and } j = 1, \ldots, n_f \tag{7.10}$$

where $\mathrm{T}_{(n_b+i),j,z}$ are the elements of the training matrix $\mathbf{T}$ associated to inner pressure residuals, obtained as described in Chapter 2, with size $(n_b + n_f) \times n_c \times m_T$ where all the instances are stored with features as rows, classes as columns and different instances (examples) in the third dimension.

It is considered that an indirect measure of the relevance of each feature, R_z for $z = 1, \ldots, n_f$, can be computed as follows

$$R_z = \frac{2}{n_f^2 - n_f} \sum_{i=1}^{n_f-1} \sum_{j=i+1}^{n_f} (\Lambda_{z,i} - \Lambda_{z,j})^2 \tag{7.11}$$

Redundancy metric

As an indirect measure of the redundancy of $\mathbf{F}$, the similarity or proximity degree between each pair of features is used. In (Sarrate et al., 2014a), it was proposed to use row vectors of the leak sensitivity matrix to measure the similarity between the behavior of two inner pressure sensors in the presence of the different leak scenarios. In this work, we propose to use the 2-norm between the average values of each possible pair of features. Then, considering the row vectors of the matrix

$$\mathbf{\Lambda} = \begin{pmatrix} \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_{n_f} \end{pmatrix} \tag{7.12}$$

where

$$\boldsymbol{\lambda}_i = (\Lambda_{i,1}, \ldots, \Lambda_{i,n_c}) \text{ for } i = 1, \ldots, n_f \tag{7.13}$$

A feature distance matrix $\boldsymbol{\Phi}$ can be defined such that its components store the measure of the redundancy between each pair of features i and j and are computed as

$$\Phi_{i,j} = \|\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_j\|_2 \text{ for } i = 1, \ldots, n_f \text{ and } j = 1, \ldots, n_f \tag{7.14}$$

where the matrix $\boldsymbol{\Phi}$ is a symmetric square matrix of dimension n_f where all the diagonal elements are 0, thus indicating that each feature presents zero distance to itself, in other words, each feature is totally redundant to itself.

Filtering process

The filtering process starts by computing the relevance for all the features, R_z for $z = 1, \ldots, n_f$, according to (7.11). The computed values are introduced in the relevance vector $\mathbf{R}$ and they are sorted in descending order in $\mathbf{R}_R$. On the other hand, the feature distance matrix $\boldsymbol{\Phi}$, that stores the distance between pairs of features is computed, according to (7.14).

The core of the algorithm is an iterative process that starts by considering the feature corresponding to the first value of $\mathbf{R}_R$, i.e. the most relevant feature. This feature is first compared in terms of similarity with the next feature in the relevance ranking. Taking into account the associated coefficient in $\boldsymbol{\Phi}$ and a user defined threshold γ, if the distance between the two considered features is lower than the threshold then the second (and less relevant) feature is removed from the feature space $\mathbf{F}$. The comparison and elimination process is repeated until the most relevant feature has been compared with all the other features in the list. And the whole process already applied to the first feature is repeated for the rest of features in the list. At the end, the feature space with the remaining features $\mathbf{F}^{(R)}$ is obtained, being its dimension $n_f^{(R)}$.

Finally, the filtering process ends with the computation of a matrix that will be used

in the wrapper stage. According to a new user defined threshold φ, a binary square matrix $\mathbf{\Phi}^{(B)}$ of dimension $n_f^{(R)}$ is defined. This matrix collects the information of which pairs of features of the remaining $n_f^{(R)}$ features after the filtering stage are suitable to be combined or not in the same potential feature selection group in the wrapping stage according their dissimilarity. The components of this matrix are computed as

$$\Phi_{i,j}^{(B)} = \begin{cases} 0 & \text{if } \left|\Phi_{i',j'}\right| < \varphi \\ 1 & \text{if } \left|\Phi_{i',j'}\right| \geq \varphi \end{cases} \tag{7.15}$$

where the indices i and j are related with the indices i' and j' with a mapping function that maps the features of $\mathbf{F}^{(R)}$ in the features of the original feature space $\mathbf{F}$.

Notice that φ has to be bigger than γ to have an impact on the wrapper because pairs of features with feature distance smaller than γ are removed in the filtering stage. Both values for φ and γ can be expressed in a relative way with respect to the average of the coefficients of $\mathbf{\Phi}$ outside the main diagonal, denoted as $\mathfrak{y}$ and computed as

$$\mathfrak{y} = \frac{2}{n_f^2 - n_f} \sum_{i=1}^{n_f-1} \sum_{j=i+1}^{n_f} \Phi_{i,j} \tag{7.16}$$

The whole filter process is summarized in Algorithm 2.

Wrapper search

The wrapper used in the second stage of the hybrid feature selection proposed in this method is a genetic algorithm.

The two modifications to the genetic algorithm proposed in the previous sensor placement technique are also applied here.

A new training matrix $\mathbf{T}^{(R)}$ is built by removing the features discarded by the filter. This matrix has dimension $(n_b + n_f^{(R)}) \times n_c \times m_T$. In a similar way, a new validation matrix $\mathbf{V}^{(R)}$ of $(n_b + n_f^{(R)}) \times n_c \times m_V$ dimension is created, where m_V is the number

Algorithm 2 Relevance and redundancy/distance filter.

Require: A features space $\mathbf{F}$ and their size n_f, a training matrix $\mathbf{T}$, the number of classes $n_c = n_f$ of each feature (n_f is used instead of n_c), the number of instances in each class m_T in the training matrix and the user defined thresholds γ and φ.

Ensure: Remove the redundant (and less relevant) features below $\mathfrak{H}$.

for $i = 1, \ldots, n_f$ **do**
  for $j = 1, \ldots, n_f$ **do**
    $\Lambda_{i,j} = \frac{\mathbf{1}}{\mathbf{m_T}} \sum_{a=1}^{m_T} \mathrm{T}_{(n_b+i),j,a}$
  end for
end for
for $z = 1, \ldots, n_f$ **do**
  $R_z = \frac{2}{n_f^2 - n_f} \sum_{i=1}^{n_f-1} \sum_{j=i+1}^{n_f} (\Lambda_{z,i} - \Lambda_{z,j})^2$
end for
Rank in $\mathbf{R}_R$ the features according to their value in $\mathbf{R}$.
for $i = 1, \ldots, n_f$ **do**
  for $j = 1, \ldots, n_f$ **do**
    $\Phi_{i,j} = \|\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_j\|_2$
  end for
end for
for $i = 1, \ldots, n_f$ **do**
  if $R_{R,i} \geq 0$ **then**
    for $j = i + 1, \ldots, n_f$ **do**
      if $\Phi_{i,j} < \gamma$ **and** $R_{R,j} \geq 0$ **then**
        $R_{R,j} = -1$
      end if
    end for
  end if
end for
Removed all the features from the space $\mathbf{F}$ with negative argument in $\mathbf{R}_R$ to create the new reduced space $\mathbf{F}^{(R)}$ with $n_f^{(R)}$ number of features. Also create, using φ and (7.15), the binarized feature distance matrix $\mathbf{\Phi}^{(B)}$.

of instances of each class in the validation data set.

The pseudo-code of the algorithm is shown in Algorithm 3. First, the GA is initialized by adjusting the tuning parameters (line 1) which include the population size p_s, the bit string type population, the elite count e_c to maintain members between iterations, the tolerance *tol* and the maximum number of generations max_g to stop the optimization. Then, the constraints of the optimization variables are defined (line 2), which include the number of features to select n_s.

After that, the GA optimization process runs as an iterative process (lines 4 to 24), where the first step is to create the generation of members ($\mathbf{I}$ matrix) to be evaluated (line 6) and then evaluate them all (lines 7 to 22). Firstly, the algorithm checks (function GetUsed()) if the member ($\mathbf{q}$ vector) is new or repeated (line 9). If the combination is repeated the stored fitness value is retrieved (function GetATD()) (line 20) instead of computing the fitness function again, and the member evaluation finalizes, otherwise the process continues. The next step in the process is to check if the member is an allowed combination of features according to the $\mathbf{\Phi}^{(B)}$ matrix (function CheckCombinations()) (line 10). If the combination is not allowed, then the fitness value is set to the worst result (line 17) and the member evaluation finalizes; otherwise, if it is allowed, then the process continues.

To perform the evaluation itself, first the classifier is created by using the training matrix $\mathbf{T}^{(R)}$ (line 11) where the $\mathbf{q}_t$ vector is used to select the adequate columns, and the confusion matrix $\mathbf{\Gamma}$ is obtained by using the classifier and the validation matrix $\mathbf{V}^{(R)}$ (line 12). Then, the fitness indicator is computed from the confusion matrix (line 13), the member is set to used (function SetUsed()) (line 14) and the fitness value is stored (function SetATD()) (line 15).

Finally, the best member of the generation is evaluated against the past ones, and it is checked if any of the required criteria to stop the optimization is reached (line 23).

Algorithm 3 Sensor placement based on Genetic Algorithms.

Require: A training matrix $\mathbf{T}^{(R)}$ and a validation matrix $\mathbf{V}^{(R)}$. A feature space $\mathbf{F}^{(R)}$ and their size $n_f^{(R)}$, the number of classes $n_c = n_f$, the number of features to select n_s, the population size p_s, the elite count parameter e_c, the fitness function tolerance tol, the maximum number of generations allowed max_g, the distance matrix $\mathbf{D}$ and the reduced binarized matrix $\mathbf{\Phi}^{(B)}$.

Ensure: A near-optimal sensors configuration $\mathbf{q}$ with error index $\text{ATD}_{\mathbf{min}}$.

$init \leftarrow \text{InitVarGA}(p_s, e_c, tol, max_g)$
$constraint \leftarrow \text{SetConstraints}(n_f^{(R)}, n_s)$
Inputs: *init*, *constraint*, $\mathbf{T}^{(R)}$, $\mathbf{V}^{(R)}$, p_s, n_c, $\mathbf{\Phi}^{(B)}$.
while An optimization criterion is not reached **do**
  GA based search:
  Generate $\mathbf{I}$ matrix of size $(p_s \times n_f^{(R)})$ where each row is a member of a generation from the space $\mathbf{F}^{(R)}$.
  $\mathbf{Q} = \mathbf{I}$
  for $t = 1, \ldots, p_s$ **do**
    if $\text{GetUsed}(\mathbf{q}_t) = 0$ **then**
      if $\text{CheckCombinations}(\mathbf{\Phi}^{(B)}, \mathbf{q}_t) = 1$ **then**
        $\text{Classifier}(\mathbf{q}_t) \leftarrow \text{Train}(\mathbf{T}^{(R)}(\mathbf{q}_t))$
        $\mathbf{\Gamma}(\mathbf{q}_t) \leftarrow \text{Validate}(\text{Classifier}(\mathbf{q}_t), \mathbf{V}^{(R)}(\mathbf{q}_t))$
        $\text{ATD}(\mathbf{q}_t) \leftarrow \frac{\sum_{i=1}^{n_f}\sum_{j=1}^{n_f}\Gamma_{i,j}(\mathbf{q})\text{D}_{i,j}}{\sum_{i=1}^{n_f}\sum_{j=1}^{n_f}\Gamma_{i,j}(\mathbf{q}_t)}$
        $\text{SetUsed}(\mathbf{q}_t)$
        $\text{SetATD}(\text{ATD}(\mathbf{q}_t), \mathbf{q}_t))$
      else
        $\text{ATD}(\mathbf{q}_t) = 0$
      end if
    else
      $\text{ATD}(\mathbf{q}_t) = \text{GetATD}(\mathbf{q}_t)$
    end if
  end for
  Find $\{\mathbf{q}, \text{ATD}_{\mathbf{min}}\}$ such that
    $\text{ATD}_{\mathbf{min}} = \min_{\mathbf{q}}(\text{ATD}(\mathbf{q}_1), \ldots, \text{ATD}(\mathbf{q}_{p_s}))$.
end while

7.2. Incremental Feature Selection Approach

The proposed sensor placement methods presented so far have a good performance when the number of sensors is low. But when the number of sensors n_s is large there are more efficient tools to deal with the problem. In this section a sensor placement for a large number of sensors is presented but focused to suit with the data-driven leak localization approach presented in Chapter 6. Therefore, an indicator to deal with the specific problem of sensor placement in the data-driven leak localization approach is introduced.

As stated before, for a given a fixed number n_s of sensors to be placed in a WDN, the optimal sensor placement can be formulated as an optimization problem with binary decision variables as follows

$$\min_{\mathbf{q}} e(\mathbf{q})$$

$$\text{s.t.} \tag{7.17}$$

$$\sum_{i=1}^{n_f} \mathrm{q}_i = n_s$$

where $\mathbf{q}$ is the binary vector that characterizes the locations of the sensors defined in (7.1) and e is a cost function to be minimized. The e cost function is made with the aim of obtain the best fitting by the interpolation technique. In this case, the Kriging interpolation is used but must be noted that any other interpolation technique is applicable with the sensor placement presented. The sum of squares relative errors of the interpolation in pressure values is used as cost function e considering non-leak and leak scenarios under different operational conditions $\mathbf{c}_d$ is computed as

$$e(\mathbf{q}) = \sum_{j=0}^{n_n} \sum_{z=1}^{m_T/(n_n+1)} \sum_{i=1}^{n_n} \left(\frac{p_i^{(l_j)}(\mathbf{c}_d(z)) - \hat{p}_i^{(l_j)}(\mathbf{c}_d(z), \mathbf{q})}{p_i^{(l_j)}(\mathbf{c}_d(z))} \right)^2 \tag{7.18}$$

where l_0 denotes no-leak scenario, $m_T/(n_n+1)$ denotes the number of data samples in no-leak scenario and in the different leak scenarios and $\hat{p}_i^{(l_j)}(\mathbf{c}_d, \mathbf{q})$ denotes pressure estimations using sensor configuration defined by $\mathbf{q}$. To generate the data required here, measurements from all nodes and measurements with leaks in all the nodes, a hydraulic simulator and the scheme depicted in Figure 7.4 is used.

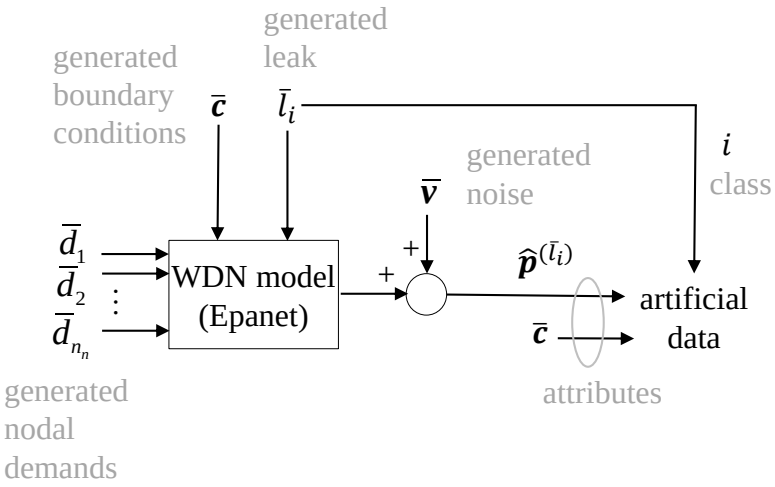


Figure 7.4: Pressure data generation scheme.

As a remark, the accuracy of the hydraulic simulator necessary to generate the data for the sensor placement problem is not as critical as if the hydraulic simulator was used in a model-based leak localization scheme. The purpose of the hydraulic simulator that generates pressure values in (7.18) is to have an idea about the pressure map in the WDN in order to determine the optimal placement of the pressure sensors by means of the optimization problem (7.17).

In this case, the sequential forward floating search algorithm presented in (Pudil et al., 1994) and described in Section 2.4 is proposed to solve the optimization problem (7.17) in a suboptimal but efficient way. The particularization of this algorithm to solve the optimization problem (7.17) with cost function (7.18) is described in Algorithm 4.

This algorithm requires the number of sensors to be installed n_s and a matrix $\mathbf{T}$

where pressure simulation values $p_i^{(l_j)}(\mathbf{c}_d(z))$ for $i = 1, \ldots, n_n$, $j = 0, \ldots, n_n$ and $z = 0, \ldots, m_T/(n_n + 1)$ are stored. The pressure data is arranged as described in Figure 7.2 where the added non-leak case is considered to stay in the zero index of the second dimension. On the other hand, the algorithm returns a binary matrix $\mathbf{Q}^{(H)} \in \{0,1\}^{n_s} \times \{0,1\}^{n_f}$ that stores the optimal configurations for an incremental number of sensors $n_{sp} = 1, \ldots, n_s$ in its rows $\mathbf{q}_{n_{sp}}^{(H)} \in \{0,1\}^{n_f}$.

First, the variable corresponding to the number of sensors already placed n_{sp} is initialized to 0 (line 1), binary matrix $\mathbf{Q}^{(H)}$ is set to zero and a vector $\mathbf{e}^{(H)} \in \Re^{n_{sp}}$ whose components $e_{n_{sp}}$ will store cost function associated with sensor configuration $\mathbf{q}_{n_{sp}}^{(H)}$ is also set to zero. In addition, an auxiliary vector $\mathbf{q} \in \{0,1\}^{n_f}$ to store a sensor configuration is set to zero (no sensor selected). Then, the algorithm to place a given number of sensors is executed (lines 2-51) until $n_{sp} = n_s$ at the end of an iteration is accomplished. The Algorithm 4 is divided in two parts: the forward part and the backward part.

The forward part (lines 3-23) consists in adding the new feature that taking into account the features already selected $\mathbf{q}$, minimizes the cost function (7.18). For this purpose, a vector $\mathbf{e} \in \Re^{n_n}$ stores in its components e_i the cost function (7.18) of previous sensor configuration and adding sensor i. The computation of cost function (7.18) is carried out in lines 7-13. Kriging$(\mathbf{q}, \mathbf{t}_z)$ denotes the Kriging interpolation for all the pressure values using the measurements associated to variable $\mathbf{q}$. If sensor i is already placed, cost function (7.18) is not computed and component e_i is set to ∞ (line 16) in order not to consider this impossible case (adding an existing sensor). Once the n_f cost functions corresponding to the different n_f possible sensors have been computed, the sensor addition that produces a minimum cost value is selected (lines 19-20) and the sensor combination $\mathbf{q}$ is stored in $\mathbf{q}_{n_{sp}}^{(H)}$, i.e., ${n_{sp}}^{th}$ row of matrix $\mathbf{Q}^{(H)}$ (line 22). In addition, the value of the cost function of the selected sensor configuration is stored in component ${n_{sp}}^{th}$ of vector $\mathbf{e}^{(H)}$ (variable $e_{n_{sp}}^{(H)}$ in line 23). The main advantage of the forward part is that sensors are added sequentially, therefore the computational cost is linear with the possible number of place the sensors. However, it is not guaranteed that the obtained solution is the best solution

among the all possible sensor combinations. In order to minimize the effect of the suboptimality solution of the forward part, a backward is performed.

The backward part (lines 24-50) consists in given a sensor configuration $\mathbf{q}$ with n_{sp} selected sensors, the different n_{sp} sensor configurations of $n_{sp}-1$ sensors obtained by subtracting one sensor of the original sensor configuration $\mathbf{q}$ are evaluated. If the cost function of these configuration is smaller than the one stored in $e^{(H)}_{n_{sp}-1}$ the obtained sensor configuration is stored in $(n_{sp}-1)^{th}$ row of matrix $\mathbf{Q}^{(H)}$ (variable $\mathbf{q}^{(H)}_{n_{sp}-1}$ in line 44) and the number of selected sensors is set to $n_{sp}-1$. The backward part is executed after the forward part if the number of selected sensors is more than two and is executed until there is not an improvement in the new sensor configurations or the number of sensors becomes equal to two.

7.3. Case Studies

The proposed sensor placement approaches for the classifier-based leak localization presented in Chapter 5 are tested in two different networks, while the sensor placement for the data-driven approach presented in Chapter 6 is tested in one of them. 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 (MATLAB, 2015). The sensor placement approaches that use genetic algorithms are coded using the Matlab 2015a Global Optimization Toolbox software, for the considered case studies, considering the following parameters:

- Tolerance of $tol = 10^{-6}$.
- Population size $p_s = 5$.

Algorithm 4 Sequential forward floating search for sensor placement.

Require: The number of sensor to place n_s, the training data set $\mathbf{T}$ and the number of samples in the training data set m_T.

Ensure: Suboptimal sensor placement for multiple number of sensors starting from 1 and ending in n_s in matrix $\mathbf{Q}^{(H)}$.

$n_{sp} = 0$, $\mathbf{Q}^{(H)} = \mathbf{0}$, $\mathbf{q} = \mathbf{0}$, $\mathbf{e}^{(H)} = \mathbf{0}$
while $n_{sp} < n_s$ **do**
  for $i = 1, \ldots, n_f$ **do**
    $e_i = 0$
    if $q_i = 0$ **then**
      $q_i = 1$
      for $j = 0, \ldots, n_n$ **do**
        for $z = 1, \ldots, m_T/(n_n + 1)$ **do**
          $\mathbf{p} = \mathbf{T}_{:,j,z}$
          $\hat{\mathbf{p}} = \text{Kriging}(\mathbf{q}, \mathbf{p})$
          $e_i = e_i + \sum_{k=1}^{n_n} \left(\frac{p_k - \hat{p}_k}{p_k} \right)^2$
        end for
      end for
      $q_i = 0$
    else
      $e_i = \infty$
    end if
  end for
  $s = \text{argmin}(\mathbf{e})$
  $q_s = 1$
  $n_{sp} = n_{sp} + 1$
  $\mathbf{q}^{(H)}_{n_{sp}} = \mathbf{q}$
  $e^{(H)}_{n_{sp}} = \min(\mathbf{e})$
  while $n_{sp} > 2$ **do**
    for $i = 1, \cdots, n_f$ **do**
      $e_i = 0$
      if $q_i = 1$ **then**
        $q_i = 0$
        for $j = 0, \ldots, n_n$ **do**
          for $z = 1, \ldots, m_T/(n_n + 1)$ **do**
            $\mathbf{p} = \mathbf{T}_{:,j,z}$
            $\hat{\mathbf{p}} = \text{Kriging}(\mathbf{q}, \mathbf{p})$
            $e_i = e_i + \sum_{k=1}^{n_n} \left(\frac{p_k - \hat{p}_k}{p_k} \right)^2$
          end for
        end for
        $q_i = 1$
      else
        $e_i = \infty$
      end if
    end for
    if $e^{(H)}_{n_{sp}-1} > \min(\mathbf{e})$ **then**
      $s = \text{argmin}(\mathbf{e})$
      $q_s = 0$
      $\mathbf{q}^{(H)}_{n_{sp}-1} = \mathbf{q}$
      $e^{(H)}_{n_{sp}-1} = \min(\mathbf{e})$
      $n_{sp} = n_{sp} - 1$
    else
      Break while.
    end if
  end while
end while

- Elite count of $e_c = 0.05p_s$, but at least one ($e_c = 1$) survives (which is the case given the p_s selected).
- The maximum number of generations is $max_g = 50$.

The other parameters of the GA implementation used here in Matlab are left by their default value, which include a tournament selection value of 2, a Laplace crossover of 0.8 and a power mutation of 0.1.

7.3.1. Hanoi WDN Case Study

The Hanoi (Vietnam) WDN, presented in Chapter 3, is a simplified network of the real one, and consists of one reservoir, 31 consumer nodes and 34 pipes. The water consumption has a daily pattern similar to the one depicted in Figure 5.7 (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 ± 12.5 %). Given the size of the network, it is considered the placement of only two pressure sensors as presented by (Casillas et al., 2013). These sensors, and the flow sensor at the inlet are considered to operate with a sampling time of 10 minutes.
For the simplified Hanoi WDN, the Exhaustive Search (ES) method can be applied. The ES method guarantees the optimal solution by performing all the possible combinations among all the features, which is $\frac{n_f!}{n_s!\,(n_f - n_s)!}$. Thus, this method could suitable in terms of computational time. Given the small size of the Hanoi WDN, and the constraint of only two sensors to be selected (n_s=2) among 31 possible places where pressure sensors can be installed (n_f=31), the ES method can be applied because the $\left(\frac{31!}{2!\,(31-2)!}\right) = 465$ possible combinations is a reasonable number to be evaluated exhaustively.
To generate the data sets, three different uncertainty sources are considered in the following way:

- The demand uncertainty source has a magnitude of ± 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 ± 5 % of the average value of all pressure residuals.

Sensor Placement Using Genetic Algorithms

Using all of these uncertainty levels, ten complete data sets (since the lower the data set size, the faster the computation is and the GAs need to be executed several times to avoid local minima, changing the data set allows to avoid strange, due to the outliers, data sets) have been created simulating the pressure measurements at each possible sensor location and simulating leaks at each potential leak location (class) where data are generated with a sampling time of 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 sensitivity matrix (7.4), one instance is generated for each class and sensor (complete sensitivity matrix) with a value of total consumption of water of 2991.1 [l/s] and leak size of 50 [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 $\mathbf{\Phi}^{(B)}$ matrix (where it is used a φ value of the average value of all the $\mathbf{\Phi}$ except the diagonal, i.e., $\varphi = \mathfrak{y}$) 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 7.1 and Table 7.2, and for the case of the Bayesian classifier, where PDFs are calibrated assuming Gaussian distribution (as justified in Chapter 5 by the one-dimension Kolmogorov-Smirnov test), in Table 7.3 and Table 7.4. The genetic algorithm is 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 7.1: Sensor placement results using ES and GA in the Hanoi WDN for the k-NN classifier.

Data set	Exhaustive search			Genetic algorithm		
	Sensors	Time	Ac	Sensors	Time	Ac
1	14, 27	474	39.11	9, 15	71	31.19
2	14, 29	471	38.14	14, 29	70	38.14
3	14, 28	470	38.68	14, 28	47	38.68
4	**14, 28**	**499**	**41.06**	**14, 28**	**44**	**41.06**
5	14, 28	472	39.03	1, 30	14	16.80
6	14, 27	473	38.02	10, 15	45	32.58
7	15, 28	473	38.52	15, 28	55	38.52
8	15, 28	477	38.02	26, 28	29	35.55
9	14, 27	469	38.89	5, 14	50	31.16
10	14, 28	474	38.91	15, 29	51	36.88
Average	-	475	38.83	-	47	34.05

Table 7.2: Sensor placement results using GA + $\mathbf{\Phi}^{(B)}$ in the Hanoi WDN for the k-NN classifier.

Data set	Genetic algorithm + $\mathbf{\Phi}^{(B)}$			
	Sensors	Time filter	Time GA	Ac
1	14, 27	0.06	63	39.11
2	14, 29	0.06	38	38.68
3	14, 31	0.06	19	34.34
4	**14, 28**	**0.06**	**33**	**41.06**
5	14, 28	0.06	46	39.03
6	14, 27	0.06	36	36.07
7	15, 28	0.06	43	38.52
8	15, 28	0.06	23	38.02
9	14, 29	0.06	36	37.56
10	4, 15	0.06	15	29.04
Average	-	0.06	35	37.14

From these results, it can be seen that both methods present an important improvement in terms of computational time when GAs are used, and the GA standalone

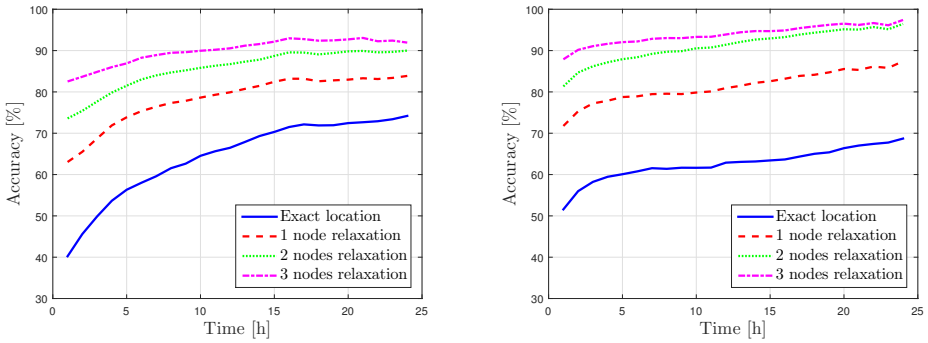
Table 7.3: Sensor placement results using the ES and GA in the Hanoi WDN for the Bayesian classifier.

Data set	Exhaustive search			Genetic algorithm		
	Sensors	Time	Ac	Sensors	Time	Ac
1	14, 28	537	51.57	9, 15	71	42.00
2	14, 28	535	51.65	14, 29	69	51.65
3	14, 28	544	52.01	14, 28	47	52.01
4	**14, 28**	**545**	**52.55**	**14, 28**	**44**	**52.55**
5	14, 28	578	51.41	1, 30	13	46.47
6	26, 27	583	46.72	10, 15	44	43.99
7	14, 28	537	52.12	14, 28	85	52.12
8	13, 28	536	47.33	13, 28	55	47.33
9	14, 28	599	52.37	5, 14	50	43.56
10	15, 28	535	46.92	6, 26	40	37.29
Average	-	553	50.46	-	58	46.90

Table 7.4: Sensor placement results using GA + $\mathbf{\Phi}^{(B)}$ in the Hanoi WDN for the Bayesian classifier.

Data set	Genetic algorithm + $\mathbf{\Phi}^{(B)}$			
	Sensors	Time filter	Time GA	Ac
1	14, 28	0.06	47	51.57
2	14, 28	0.06	57	51.65
3	14, 28	0.06	55	52.01
4	**14, 28**	**0.06**	**51**	**52.55**
5	4, 13	0.06	29	35.53
6	7, 28	0.06	32	40.13
7	14, 29	0.06	72	50.73
8	13, 28	0.06	62	47.33
9	14, 28	0.06	53	52.37
10	15, 30	0.06	50	46.57
Average	-	0.06	51	48.04

method and the GA plus $\mathbf{\Phi}^{(B)}$ are able to avoid the local minima and find the global optima in some cases (the best sensor placements obtained). Moreover, note that in average the introduction of the $\mathbf{\Phi}^{(B)}$ 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 $\mathbf{\Phi}^{(B)}$ 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 (as proposed in Chapter 5 and Chapter 6) in Figure 7.5a for the k-NN classifier (with $k = 1$) and in Figure 7.5b 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.



(a) Accuracy curves for the k-NN classifier in the Hanoi WDN with sensor placement at nodes 14 and 28.

(b) Accuracy curves for the Bayesian classifier in the Hanoi WDN with sensor placement at nodes 14 and 28.

Figure 7.5: Accuracy curves for both classifiers using the sensor placements obtained using GA with the objective to maximize the Ac in Hanoi WDN.

The results in this network show that the best performance is achieved with the Bayesian classifier being in agreement with the results presented in Chapter 5. The

sensor placement results can be seen in Figure 7.6.

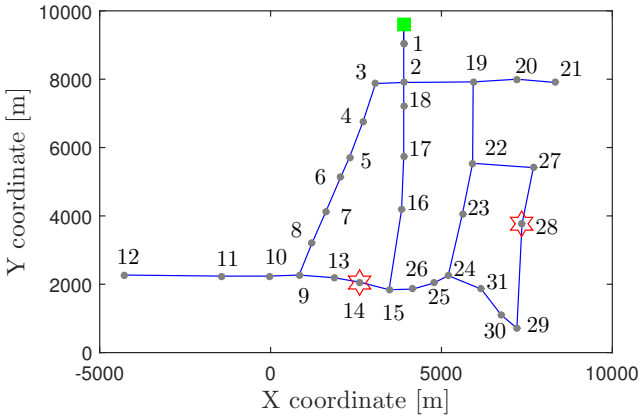


Figure 7.6: Sensor placement for the k-NN and Bayesian classifiers (same sensors) in Hanoi WDN.

Hybrid Feature Selection Approach

The k-NN classifier and the Bayesian classifier are fed with the training matrix $\mathbf{T}^{(R)}$ to store the data in the case of the k-NN classifier and to calibrate the PDFs for the Bayesian classifier, which are considered to have Gaussian distribution. Ten different data sets (training and validation) are created to evaluate the different methods. Each data set (note that the data sets are not the same as the ones used in the previous GA approach) is split into a training and validation data subsets. The training data subset consists in four days of data (96 samples for each class) and the validation data subset consists in two days of data (48 samples for each class). The training data subset is used in the filter stage to generate the subset of features $\mathbf{F}^{(R)}$ and in the wrapper stage to train the classifier for the current feature selection candidate. On the other hand, the validation data subset is used to assess the performance (ATD value) of the current feature selection candidate produced by the wrapper. To compare the different results obtained for the different data sets, a unique testing data set is used. This testing data set consists in 20 days of data (480 samples for each class) and is used to calculate the ATD value showed in Table 7.5 and Table 7.7 for the k-NN classifier with $k = 1$ and in Table 7.6 and Table 7.8 for

Table 7.5: Results of the ES and GA methods in the Hanoi WDN for the k-NN classifier.

Data set	Exhaustive search			Genetic algorithm		
	Sensors	Time ES	ATD	Time GA	ATD	
1	10, 15	607	1.66	10, 29	69	1.57
2	11, 29	579	1.53	11, 27	38	1.55
3	13, 27	591	1.50	10, 15	71	1.68
4	13, 27	565	1.39	**13, 27**	**100**	**1.39**
5	13, 29	554	1.43	13, 31	67	1.52
6	13, 29	566	1.44	11, 15	87	1.62
7	13, 29	585	1.51	10, 29	66	1.58
8	13, 29	568	1.49	26, 30	34	2.20
9	14, 27	554	1.62	13, 30	34	1.46
10	**13, 28**	**545**	**1.37**	10, 28	53	1.41
Average	-	572	1.49	-	62	1.60

the Bayesian classifier.

The results obtained by the ES method are used to compare the efficiency of the final solution and computation time for the proposed approach. On the other hand, a standalone GA wrapper method that only considers the proposed wrapper stage (Algorithm 3 without previous filtering nor the use of the matrix $\mathbf{\Phi}^{(B)}$) will be applied to illustrate the advantages in the sensor placement performance of the proposed hybrid method (filter+wrapper).

The results for the ES method in the Hanoi WDN are summarized in Table 7.5 and Table 7.6 for the k-NN and the Bayesian classifiers respectively, where the term "Sensors" refers to the selected node locations where the inner pressure sensors will be placed, "ATD" is the average topological distance in [nodes] computed by (2.17) using the testing data set, and "Time ES" is the time required to obtain the solution by the ES method in [s].

It can be noted from Table 7.5 and Table 7.6 that different data sets lead to different results, i.e., to different pairs of selected sensors, as it happens in the previous method. This is due to the fact that there are sets of close nodes for which the behavior in terms of pressure is quite similar and due to the randomness of the data generation. Then, the results for one data set can include a given feature/node

Table 7.6: Results of the ES and GA methods in the Hanoi WDN for the Bayesian classifier.

Data set	Exhaustive search			Genetic algorithm		
	Sensors	Time ES	ATD	Time GA	ATD	
1	14, 28	573	1.21	14, 28	111	1.21
2	11, 28	570	1.24	11, 29	60	1.26
3	10, 27	579	1.13	9, 15	49	1.39
4	**13, 27**	**569**	**1.09**	26, 27	35	1.62
5	13, 27	570	1.11	10, 27	57	1.16
6	13, 27	573	1.13	11, 27	66	1.18
7	13, 29	571	1.13	10, 29	69	1.17
8	13, 27	573	1.15	**13, 27**	**42**	**1.15**
9	13, 28	568	1.14	11, 15	55	1.25
10	13, 28	563	1.13	13, 23	55	1.32
Average	-	571	1.15	-	60	1.27

while the results for a different data set can include a different but close node. For example, in Table 7.6, the nodes that appear as first selected node for all the 10 data sets are nodes 10, 11, 13 and 14, which can be identified as neighbors in Figure 5.4. And the same happens with nodes 27, 28 and 29 that appear as second selected node. The finally selected sensors are the pair of nodes 13 and 27, highlighted in bold in the table and obtained for the data set 4, because for this data set and this pair of sensors the obtained ATD is the minimal one. It must be noted that the different combinations have a very close performance and all of them are good configurations to place the sensors, far better than the discarded configurations.

To compare the performance of the proposed approach, the same experiments as in the ES are done using the the pure genetic algorithm, i.e., without the improvement of the use of $\mathbf{\Phi}^{(B)}$.

The obtained results are summarized in Table 7.5 for the k-NN classifier and in Table 7.6 for the Bayesian classifier, where "Time GA" is the time used by the standalone wrapper method to select the features in [s]. Comparing with the results obtained by exhaustive search, it can be highlighted that the selected sensors are the same pair of nodes 13 and 27 in the k-NN case and almost the same in the Bayesian classifier case, and that the average computing time is reduced from 571

Table 7.7: Results of the proposed hybrid feature selection in the Hanoi WDN for the k-NN classifier.

Data set	Filter + (Genetic algorithm + $\mathbf{\Phi}^{(B)}$)				
	Sensors	Time filter	Time GA	$n_f^{(R)}$	ATD
1	14, 22	0.032	23	21	1.72
2	11, 14	0.032	35	21	1.73
3	13, 24	0.032	52	20	1.62
4	13, 23	0.032	30	21	1.56
5	13, 25	0.032	29	20	1.61
6	13, 23	0.032	29	21	1.58
7	13, 24	0.032	46	20	1.61
8	13, 25	0.032	47	21	1.60
9	**13, 24**	**0.032**	**34**	**21**	**1.53**
10	13, 25	0.032	21	21	1.54
Average	-	0.032	35	-	1.61

to 60 seconds in the Bayesian case (572 to 62 in the k-NN case).

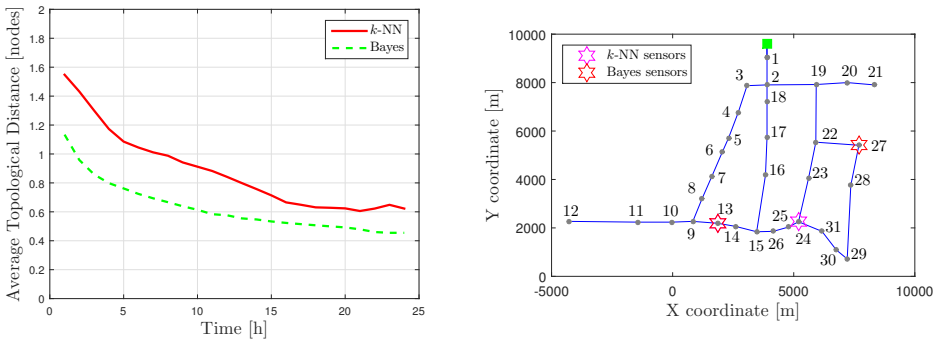
Finally, the proposed hybrid feature selection method has been tested. The values used for the thresholds has been empirically chosen according to some previous test and with the goal of banning some combinations but still allowing a rich number of tested combinations, those are $\gamma = \mathfrak{y}/2$ and $\varphi = \mathfrak{y}/4$. The obtained results are summarized in Table 7.7 (k-NN classifier) and Table 7.8 (Bayesian classifier). In these tables, "Time filter" refers to the time of filter computation in [s], "Time GA" refers to the time of wrapper computation in [s] and $n_f^{(R)}$ is the number of features that pass the filter. It can be observed that the selected pair of sensors is still the same pair of nodes for the Bayesian case while in the k-NN case is not able to reach the optimum but stays near to it and in average their accuracy is close the purely genetic algorithm method. With respect to the computing time, two aspects can be highlighted: First, the computing time of the filter is negligible with respect to the one of the wrapper (in average, 0.032 seconds versus 37 seconds). Second, the filter helps the wrapper to be faster; in particular, a decrease from 60 to 25 seconds for the k-NN case and 60 to 37 seconds for the Bayesian in the average computing time is obtained.

The ATD values in a time horizon of 24 hours is depicted in Figure 7.7a for both

Table 7.8: Results of the proposed hybrid feature selection in the Hanoi WDN for the Bayesian classifier.

Data set	Filter + (Genetic algorithm + $\mathbf{\Phi}^{(B)}$)				
	Sensors	Time filter	Time GA	$n_f^{(R)}$	ATD
1	11, 15	0.032	37	21	1.32
2	11, 29	0.032	30	21	1.27
3	13, 29	0.032	25	20	1.13
4	13, 29	0.032	33	21	1.13
5	13, 29	0.032	33	20	1.14
6	13, 27	0.032	45	21	1.13
7	10, 27	0.032	27	20	1.14
8	10, 15	0.032	34	21	1.29
9	13, 29	0.032	39	21	1.14
10	**13, 27**	**0.032**	**51**	**21**	**1.12**
Average	-	0.032	37	-	1.18

classifiers using the combination of selected sensors along with the training data set with which the best ATD value at one step is obtained and the testing data set. The resulting sensor placement for both classifiers in the Hanoi WDN is depicted in Figure 7.7b.



(a) Average topological distance for the the k-NN and the Bayesian classifier using the sensor placement obtained through the hybrid feature selection approach.

(b) Sensor placement using hybrid feature selection for the k-NN and Bayesian classifiers (both have a sensor at node 13) in Hanoi WDN.

Figure 7.7: Sensor placement and the ATD curves using the hybrid feature selection for both classifiers in Hanoi WDN.

Incremental Feature Selection Approach

Using the artificial data $\mathbf{T}$ with the uncertainty characteristics described earlier

but using and addition of white noise with amplitude of 0.1 and zero mean for the pressure measurements instead of the previous used. Using four days of data (four days with no leak and four days with leak for each consumption node, which gives a total of $m_T = 96$ samples) for the training data set, and the incremental feature selection method proposed earlier a sensor placement is obtained for a given number of sensors $n_s = 31$.

As it can be seen in the Figure 7.8, the leak localization improves with the increase of the number of sensors, specially when their number is low for then, improve in a more moderate way.

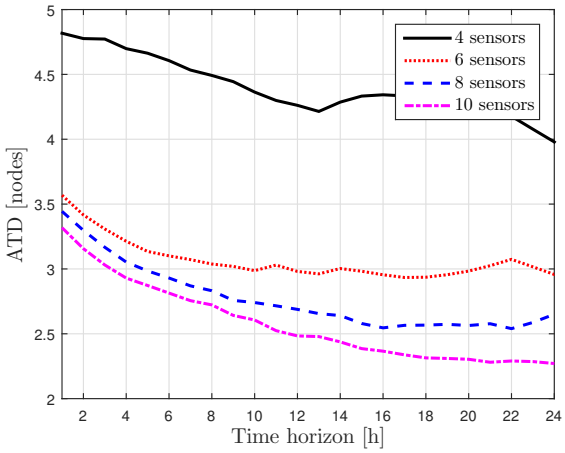
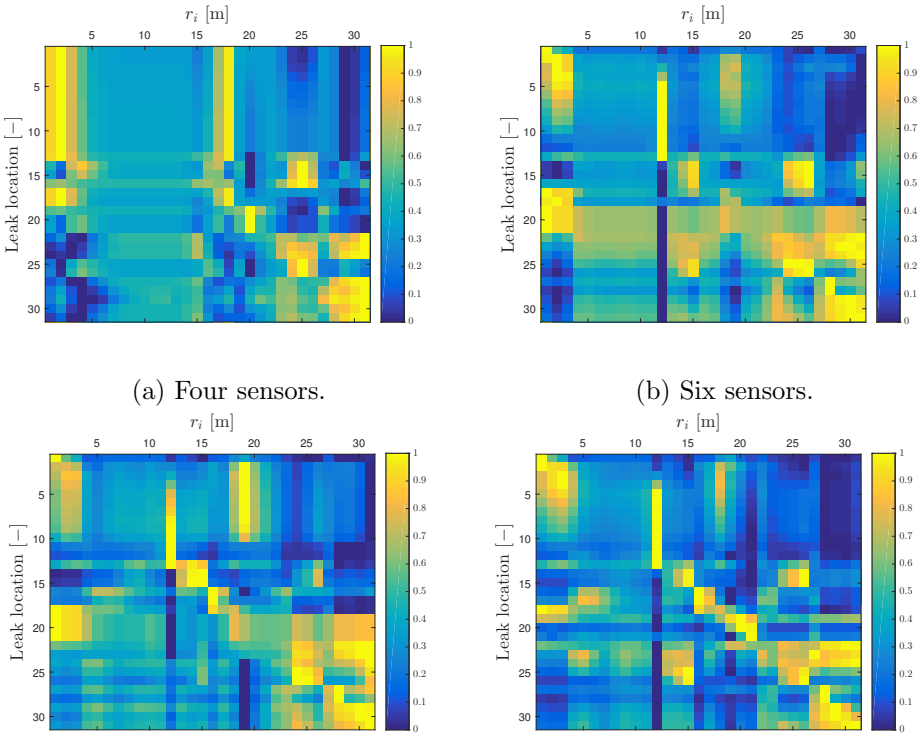


Figure 7.8: ATD performance for the incremental feature selection sensor placement using different number of sensors in the Hanoi WDN.

Similar to the Figure 6.1 where sensors in all the sensors are considered, in the Figure 7.9 the impact of lower number of sensors obtained using the proposed incremental feature selection method, can be seen in the normalized sensitivity matrix. From the results depicted, the sensor placement of six sensors is chosen since the addition of more sensors does not significantly improve. The obtained sensor placement for six sensors is shown in Figure 7.10.

It must be noted that the six sensors chosen here are not necessary the same as if the optimization had been done for $n_s = 6$. This is due to the nature of the incremental method that allows backward steps, which can improve subsets of features already



(a) Four sensors.

(b) Six sensors.

(c) Eight sensors.

(d) Ten sensors.

Figure 7.9: Normalized sensitivity matrix of the Hanoi WDN for a leak of 100 [l/s] for some sensor placements by the incremental feature selection method.

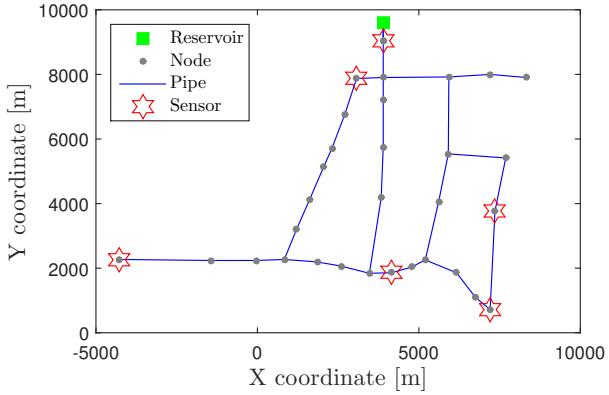


Figure 7.10: Sensor placement using incremental feature selection for six sensors in the simplified Hanoi topological WDN.

chosen. So, the use of n_s in the optimization process higher than the number of sensors that are really wanted to place could be taken into account to get better sensor configurations.

7.3.2. Limassol DMA Case Study

The Limassol DMA network described in Chapter 3 is used to test the proposed sensor placement based on genetic algorithms and the hybrid approach for both classifiers. The daily water consumption pattern (depicted in Figure 7.11) is generated as in the previous case (with different scale). For this network, it is decided to place three pressure sensors.

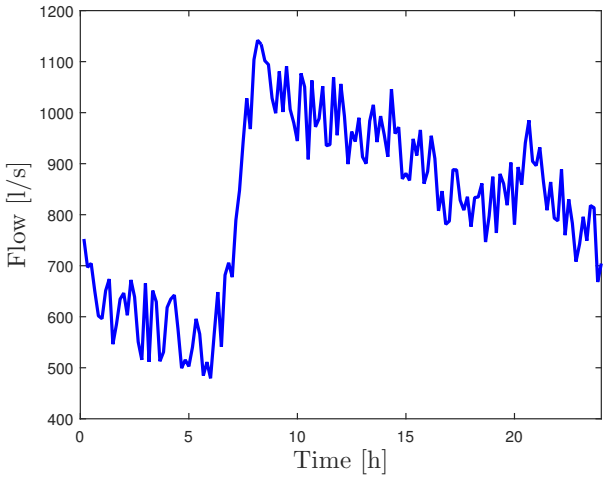


Figure 7.11: Example of a daily flow consumption in the Limassol DMA network.

For the Limassol DMA network, it is not realistic to apply the Exhaustive Search method. The total time needed to compute and evaluate the performance of all the possible combinations for 197 nodes and select three of them (3,764,670 combinations) can be roughly approximated by assuming that the time of computing each combination, estimated from the average time from 50 combinations using a Bayesian classifier, is 240.48 seconds, providing a value of 28.7 years.

Sensor Placement Using Genetic Algorithms

The data sets are generated considering similar uncertainties (in this case the leak

varies from 2 to 6 [l/s]) as the Hanoi WDN case, same sampling time (and computing the hourly average value) and the same number of examples for each class used for this method. The classifiers are built 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 7.9 and Table 7.11 for the case of the k-NN classifier ($k = 1$ as in the Hanoi WDN case), and in the Table 7.10 and Table 7.12 for the Bayesian classifier. In this case, the φ value is equal to the mean values of the $\mathbf{\Phi}$ matrix except the diagonal ($\varphi = \mathfrak{y}$). The best configurations are highlighted in bold for each method.

Table 7.9: Sensor placement results using GA in the Limassol DMA network for the k-NN classifier.

Data set	Genetic algorithm		
	Sensors	Time	Ac
1	**15, 46, 113**	**8628**	**11.03**
2	1, 7, 11	8602	10.89
3	8, 183, 195	14906	9.84
4	124, 183, 185	13957	8.09
5	3, 7, 8	5508	9.66
6	6, 8, 11	13652	10.72
7	129, 185, 190	15308	8.03
8	1, 3, 7	2010	9.07
9	87, 124, 128	12434	9.55
10	3, 166, 181	4995	7.11
Average	-	9690	9.39

In this network, as in the Hanoi WDN case, the use of the $\mathbf{\Phi}^{(B)}$ matrix reduces the computation time in most cases and in the average value. In this case the Ac value is a bit worse than the one obtained using only the GA in the average performance. 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

Table 7.10: Sensor placement results using GA in the Limassol DMA network for the Bayesian classifier.

Data set	Genetic algorithm		
	Sensors	Time	Ac
1	39, 77, 133	21348	17.35
2	7, 19, 23	18701	19.28
3	45, 110, 185	16696	16.40
4	7, 11, 110	47590	19.34
5	11, 91, 186	28032	17.43
6	39, 48, 485	28106	19.13
7	94, 133, 166	7730	19.22
8	124, 189, 192	29217	16.61
9	13, 22, 100	21904	18.86
10	**40, 66, 104**	**15221**	**20.10**
Average	-	23454	18.37

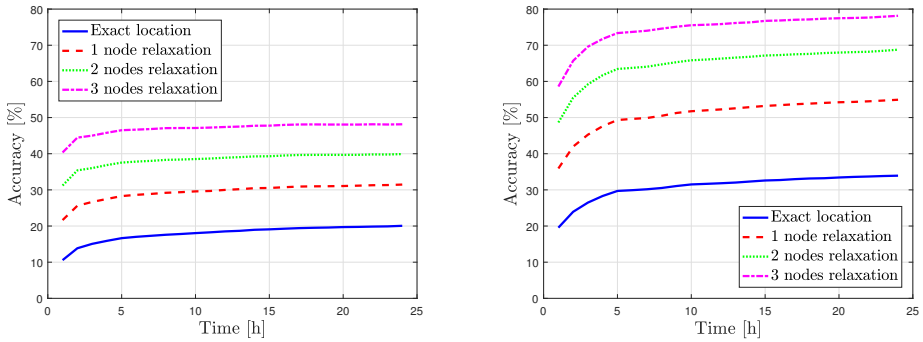
Table 7.11: Sensor placement results using GA + $\mathbf{\Phi}^{(B)}$ in the Limassol DMA network for the k-NN classifier.

Data set	Genetic algorithm + $\mathbf{\Phi}^{(B)}$			
	Sensors	Time filter	Time GA	Ac
1	1, 7, 195	6.4	7348	9.04
2	8, 102, 182	7.5	16154	10.19
3	52, 128, 133	6.8	8020	9.66
4	7, 195, 197	6.2	6580	9.66
5	1, 7, 195	7.1	1595	8.80
6	104, 183, 195	6.8	5909	8.36
7	1, 2, 197	6.3	479	5.38
8	13, 40, 167	6.3	8294	10.37
9	104, 124, 167	8.0	10551	10.37
10	**5, 11, 124**	**6.2**	**13304**	**10.39**
Average	-	6.7	7793	9.22

Table 7.12: Sensor placement results using GA + $\mathbf{\Phi}^{(B)}$ in the Limassol DMA network for the Bayesian classifier.

Data set	Genetic algorithm + $\mathbf{\Phi}^{(B)}$			
	Sensors	Time filter	Time GA	Ac
1	11, 46, 133	6.3	21255	18.90
2	17, 166, 181	6.3	17845	19.60
3	40, 75, 156	6.8	14742	15.74
4	**17, 46, 181**	**6.7**	**25515**	**19.63**
5	91, 188, 190	6.5	20117	15.67
6	39, 93, 189	6.7	15775	17.23
7	100, 124, 183	6.9	15765	18.90
8	14, 167, 185	6.4	13200	17.93
9	93, 188, 190	6.4	11453	15.97
10	13, 22, 190	6.5	6416	17.63
Average	-	6.5	16208	17.72

training data set and the testing data set) for both sensor placements is depicted in Figure 7.12a and Figure 7.12b, respectively.



(a) Accuracy curves for the k-NN classifier in the Limassol DMA network with sensor placement at nodes 5, 11 and 124.

(b) Accuracy curves for the Bayesian classifier in the Limassol DMA network with sensor placement at nodes 17, 46 and 181.

Figure 7.12: Accuracy curves for both classifiers using the sensor placements obtained using GA with the objective to maximize the Ac in Limassol DMA network.

Finally, the resulting sensor placement for the last proposed method (Genetic algorithm + $\mathbf{\Phi}^{(B)}$) for both classifiers is depicted in Figure 7.13.

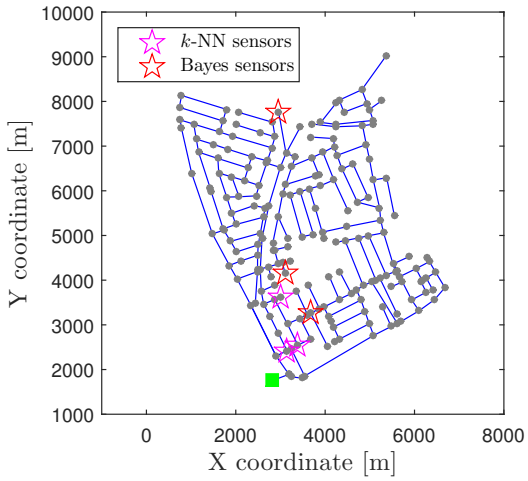


Figure 7.13: Sensor placement for the k-NN and Bayesian classifiers in the Limassol DMA network.

Hybrid Feature Selection Approach

As with the Hanoi WDN network, a simplified version of the method that just implements the wrapper has been tested first. The data sets are equal to the ones used in the Hanoi WDN case for this method and the uncertainties considered are equal to the ones used in genetic algorithm just tested. The obtained results are summarized in Table 7.13 for the k-NN classifier and Table 7.14 for the Bayesian classifier. Different combinations are selected for the different data sets, but again a further analysis shows that all the combinations are quite similar since their components are close nodes. Applied to the k-NN case, the genetic algorithm wrapper gives as a best result the set of nodes 1220, 172 and 194 an ATD of 3.9 nodes and average ATD of 4.1 nodes with an average computation time of 9,669 seconds. In the Bayesian classifier case the finally selected nodes are the set of nodes 81,133 and 169. Working with the data set 1 and using the measurements simulated for sensors in these nodes, the obtained ATD is minimal and equal to 3.06. The average value for the ATD for the 10 considered data sets is 3.39. Finally, the value for the average computation time is 18,270 seconds (around five hours).

The results for the hybrid feature selection method are summarized in the Table 7.15 and Table 7.16 for the k-NN and Bayesian classifiers respectively. The more restric-

Table 7.13: Sensor placement using only a wrapper GA method in Limassol DMA network for the k-NN classifier.

Data set	Genetic algorithm		
	Sensors	Time GA	ATD
1	34, 110, 159	15796	3.89
2	33, 111, 138	7598	4.03
3	**120, 172, 194**	**7512**	**3.9**
4	133, 185, 190	5647	4.2
5	61, 86, 189	8790	4.2
6	133, 185, 190	4945	4.2
7	103, 124, 155	13908	4.1
8	10, 47, 155	11921	4.2
9	133, 156, 196	15004	3.9
10	133, 185, 190	5569	4.2
Average	-	9669	4.1

Table 7.14: Sensor placement using only a wrapper GA method in Limassol DMA network for the Bayesian classifier.

Data set	Genetic algorithm		
	Sensors	Time GA	ATD
1	**81, 133, 169**	**19007**	**3.06**
2	10, 43, 172	24157	3.56
3	28, 110, 170	23773	3.15
4	185, 195, 196	10515	3.64
5	132, 185, 188	13481	3.18
6	87, 189, 190	18200	3.70
7	15, 51, 152	21752	3.51
8	28, 118, 156	16565	3.17
9	87, 189, 190	17526	3.70
10	12, 36, 156	17725	3.31
Average	-	18270	3.39

Table 7.15: Sensor placements of the proposed hybrid feature selection in Limassol DMA network for the k-NN classifier.

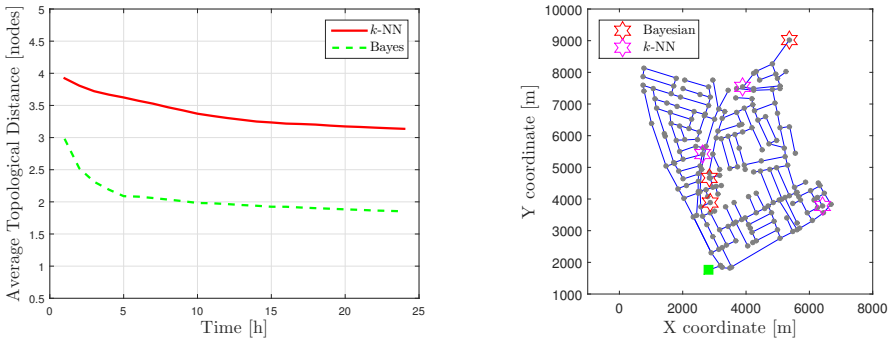
Data set	Filter + (Genetic algorithm + $\mathbf{\Phi}^{(B)}$)				
	Sensors	Time filter	Time GA	$n_f^{(R)}$	ATD
1	27, 120, 172	85	1.76	4448	3.9
2	33, 99, 155	85	1.74	3435	4.1
3	33, 133, 138	87	1.90	4916	3.9
4	68, 120, 159	87	1.97	9119	4.0
5	33, 133, 155	86	1.91	6906	3.9
6	68, 133, 151	87	1.73	12835	3.9
7	133, 159, 190	87	1.74	6281	3.9
8	120, 172, 188	87	1.60	4852	4.0
9	**33, 133, 172**	**87**	**1.71**	**6459**	**3.9**
10	29, 120, 155	87	1.63	4497	4.0
Average	-	1.77	6423	-	4.0

tive value $\varphi = \mathfrak{y}/20$ is used for this example in order to remove a suitable number of features in the filtering stage, while the $\gamma = \mathfrak{y}/2$ value is maintained. From Table 7.15 and Table 7.16, it can be seen that the proposed method performance is better in average than the standalone wrapper method in terms of the objective indicator (ATD) and with a significant reduction in the computation time.

The selected nodes to place sensors (features) have been the nodes number 33, 133 and 172 for the k-NN classifier and 40, 152 and 166 for the Bayesian classifier. The ATD values in a time horizon made as in the Hanoi WDN case are depicted in Figure 7.14a. The resulting sensor placements is depicted in Figure 7.14b.

Table 7.16: Sensor placements of the proposed hybrid feature selection in Limassol DMA network for the Bayesian classifier.

Data set	Filter + (Genetic algorithm + $\mathbf{\Phi}^{(B)}$)				
	Sensors	Time filter	Time GA	$n_f^{(R)}$	ATD
1	82, 120, 154	1.64	15828	85	3.09
2	120, 151, 188	1.61	8834	85	3.09
3	**40, 152, 166**	**1.67**	**5429**	**87**	**3.00**
4	40, 194, 197	1.57	9495	87	3.43
5	38, 120, 159	1.59	25178	86	3.02
6	126, 133, 172	1.59	7177	87	3.03
7	38, 103, 172	1.54	11786	87	3.20
8	133, 194, 197	1.90	6174	87	3.50
9	7, 154, 190	1.61	9323	87	3.19
10	40, 154, 190	1.63	12790	87	3.26
Average	-	1.65	11201	-	3.18



(a) ATD curves in Limassol network using the obtained sensor placements for both classifiers.

(b) Sensor placements using the hybrid feature selection in the Limassol DMA network for both classifiers.

Figure 7.14: Sensor placement and the ATD curves using the hybrid feature selection for both classifiers in Limassol DMA network.

8. Conclusions and Future Work

8.1. Conclusions

This PhD thesis has a whole methodology that covers the problems of leak detection, leak starting time estimation, leak size estimation, leak localization and sensor placement in WDNs.

8.1.1. Leak Detection

The proposed leak detection technique provides a highly reliable indication if leak exists or not in the WDN thanks to the validation layer developed. Moreover, the proposed approach provides the estimated time instant when the leak is originated. This information can be used to maximize the data used in the leak localization task that has to be addressed next.

Another benefit of this technique is the leak size estimation calculated in the validation layer. The main drawback of this technique is the time needed to produce the detection result.

8.1.2. Model-Based Leak Localization

From the model-based leak localization methods presented in Chapter 5 we can conclude that they present a better performance than the other techniques with the same model-based approach but without taking into account the uncertainties. It must be remarked that the use of a well calibrated hydraulic model is a key factor

of the performance of the proposed method along with the accurate estimation of the uncertainty values.

From the two classifiers tested, the Bayesian classifier has proven to be a more powerful tool to localize water leaks as long as the attributes used present a close to Gaussian distribution along with a better time reasoning. Otherwise, the use of the k-NN classifier can be a good option since produce a better adaptation to each leak shape in the residual space. The proposed methodology can be extended to other classifiers.

Finally, the limited number of sensors installed in networks makes unrealistic to expect to exactly localize the leak. So, this method has to be used to bound the zone where the workers can start to inspect the networks with devices that are able to locate the leak such as leak noise correlators or surface detectors to finally pinpoint the exact location of the water leak.

8.1.3. Data-Driven Leak Localization

The data-driven leak localization approach introduced in Chapter 6 is a new methodology that has performed really well in the real cases. This approach has the ability to be implemented in a fast and straightforward way given the limited amount of off-line task to be done.

The method is less powerful compared with the model-based leak localization methods when the hydraulic model is well calibrated. But when the hydraulic model is not well calibrated it is expected to present a better performance.

The method is very sensitive to the number of installed sensors and location. The sensors measurements should also be sensitive to variations in the pressure in all places with the aim to obtain the best pressure map to proved a good diagnostic.

8.1.4. Sensor Placement

In Chapter 7 sensor placement for both leak localization methods is successfully presented and tested.

Two sensor placements are presented for the the model-based leak localization method using classifiers. The first one follows a more traditional approach of selecting the appropriate attributes for the classifier. Here, the locations where to install pressure sensors are decided by means of an optimization problem solved using a modified genetic algorithm that provides a suboptimal solution. While this approach behaves good lacks of the improvement that the use of specialization can provide. In this case, a specific indicator to optimize as long as with relevance and redundancy metrics to maximize the sensors sensitivity at the same time that obtains different sensitivities between leaks is presented in a hybrid approach.
For the case of the data-driven leak localization method a different approach is used, where the aim is shifted to obtain the sensor configuration that provides the best accuracy representation of the pressure map to enhance the prediction of pressures. The higher number of sensors required for that leak localization method makes the use of an incremental selection rather than the use of the proposed techniques for classifier, since present a better trade-off that the absolute methods used there. It should be noted that while the leak localization method did not use hydraulic models, the sensor placement requires them.

8.2. Future Work

The proposed methods in this PhD thesis are the extension of a current line of work and the beginning of new ones. In both cases there is still room to improve as long as new technology and methods are developed.

8.2.1. Leak Detection

Further improvement time response of the leak detection can be achieved, since there are faster techniques already reported in the literature. Another problem to be addressed and not treated here is the seasonal effect which causes drifts to the total water consumption that can lead to false positives if the drift has positive slope

and omitted detections if the drift has negative slope.

8.2.2. Model-Based Leak Localization

Two classifiers are proposed and tested but many more are available in the literature including ensemble methods, i.e., different classifiers that fuse their predictions, such as boosting and bootstrapping.

Also, the future addition of automatic meter readers into water distribution networks that can allow reducing the demand uncertainty in the residual space and therefore, improve the performance of the classifiers. This approach has the drawback of needing to compute the training stage in an on-line manner according to the current demands. This is not realistic in practice for larger water distribution network unless the problem is treated through the emerging technologies of big data and cloud computing.

8.2.3. Data-Driven Leak Localization

The data-driven methods proposed here use the Kriging interpolation as a multivariate regression technique, but there are more methods that can be applied which could perform better that the one proposed.

As suggested in the previous subsection, the use of automatic meter readers can be exploited here to select a more appropriate historical measurements to perform the reference map with a more similar characteristics with the actual measurements to be compared and gain in accuracy.

To assess the pressure in new locations the use of the minimum pipe length is used but other indicators should be studied.

8.2.4. Sensor Placement

The proposed sensor placement for classifiers both make use of the modified genetic algorithm which shows greater performance but there are other heuristic methods

that can perform that part, for example, the particle swarm optimization or the firefly algorithm. Also, specific metrics developed to deal with this problem are presented in the application of the proposed algorithm but there are many others metrics that can be studied or developed.

The incremental feature selection approach makes use of an indirect indicator to be optimized in order to improve the interpolation fitting. The study of a more direct indicator regarding to the leak localization performance should be addressed. The incremental method itself can be changed for other methods that may outperform the proposed approach.

Bibliography

N. B. Adhikari. Detection of Leak Holes in Underground Drinking Water Pipelines using Acoustic and Proximity Sensing Systems. *Research Journal of Engineering Sciences*, 3(9):1–6, 2014.

K. Aksela, M. Aksela, and R. Vahala. Urban Water Journal Leakage detection in a real distribution network using a SOM. *Urban Water Journal*, 64(March): 279–289, 2009.

C. Alippi, G. Boracchi, V. Puig, and M. Roveri. An ensemble approach to estimate the fault-time instant. *Proceedings of the 2013 International Conference on Intelligent Control and Information Processing, ICICIP 2013*, 270428:836–841, 2013.

J. Almandoz, E. Cabrera, F. Arregui, E. Cabrera, and R. Cobacho. Leakage Assessment through Water Distribution Network Simulation. *Journal of water resources planning and management*, 131(6):458–466, 2005.

E. Alpaydin. Introduction to machine learning, 2010.

G. R. Anjana, K. R. Sheetal-Kumar, M. S. Mohan-Kumar, and B. Amrutur. A Particle Filter Based Leak Detection Technique for Water Distribution Systems. *Procedia Engineering*, 119(May 2016):28–34, 2015.

J. Apolloni, G. Leguizamón, and E. Alba. Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. *Applied Soft Computing*, 38:922–932, 2016.

M. Bakker, J. H. G. Vreeburg, M. Van De Roer, and L. C. Rietveld. Heuristic burst detection method using flow and pressure measurements. *Journal of Hydroinformatics*, 16(5):1194, 2014.

Michèle Basseville and Igor V. Nikiforov. *Detection of abrupt changes: theory and application*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. ISBN 0-13-126780-9.

S. B. M. Beck, M. D. Curren, N. D. Sims, and R. Stanway. Pipeline network features and leak detection by cross-correlation analysis of reflected waves. *Journal of Hydraulic Engineering*, 131(8):715–723, 2005.

J. Bicik, Z. Kapelan, C. Makropoulos, and D. A. Savić. Pipe burst diagnostics using evidence theory. *Journal of Hydroinformatics*, 13(4):596, 2011.

J. Blesa, V. Puig, and J. Saludes. Robust identification and fault diagnosis based on uncertain multiple input-multiple output linear parameter varying parity equations and zonotopes, 2012.

J. Blesa, F. Nejjari, and R. Sarrate. Robustness Analysis of Sensor Placement for Leak Detection and Location Under Uncertain Operating Conditions. *Procedia Engineering*, 89:1553–1560, 2014.

J. Blesa, F. Nejjari, and R. Sarrate. Robust sensor placement for leak location: analysis and design. *Journal of Hydroinformatics*, 18(1):136–148, 2016.

V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and information systems*, 34(3): 483–519, 2013.

G. Boracchi and M. Roveri. Exploiting self-similarity for change detection. *Proceedings of the International Joint Conference on Neural Networks*, pages 3339–3346, 2014.

G. Boracchi, V. Puig, and M. Roveri. A Hierarchy of Change-Point Methods for Estimating the Time Instant of Leakages in Water Distribution Networks. *Artificial Intelligence Applications and Innovations, Aiai 2013*, 412:615–624, 2013.

S. G. Buchberger and G. Nadimpalli. Leak Estimation in Water Distribution Systems by Statistical Analysis of Flow Readings. *Journal of water resources planning and management*, 130(4):321–329, 2004.

A. Candelieri, D. Conti, D. Soldi, and F. Archetti. Spectral clustering and support vector classification for localizing leakages in water distribution networks–the ice-water project approach. In *11th International Conference on Hydroinformatics*, 2014.

A. Caputo and P. M. Pelagagge. Using Neural Networks to Monitor Piping Systems. *Process Safety Progress*, 22(June):119–127, 2003.

M. V. Casillas, L. E. Garza-Castañón, and V. Puig. Extended-horizon analysis of pressure sensitivities for leak detection in water distribution networks. In *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pages 570–575. Elsevier, 2012.

M. V. Casillas, V. Puig, L. E. Garza-Castañón, and A. Rosich. Optimal Sensor Placement for Leak Location in Water Distribution Networks Using Genetic Algorithms. *Sensors*, 13(11):14984–15005, 2013.

M. V. Casillas, L. E. Garza-Castañón, V. Puig, and A. Vargas-Martinez. Leak Signature Space: An Original Representation for Robust Leak Location in Water Distribution Networks. *Water*, 7(3):1129–1148, 2015a.

M. V. Casillas, L. E. Garza-Castañón, and V. Puig. Sensor Placement for Leak Location in Water Distribution Networks using the Leak Signature Space. *IFAC-PapersOnLine*, 48(21):214–219, 2015b.

G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers and Electrical Engineering*, 40(1):16–28, 2014.

A. F. Colombo, P. Lee, and B. W. Karney. A selective literature review of transient-based leak detection methods. *Journal of Hydro-Environment Research*, 2(4): 212–227, 2009.

D. I. C. Covas and H. M. Ramos. Case Studies of Leak Detection and Location in Water Pipe Systems by Inverse Transient Analysis. *Journal of Water Resources Planning and Management*, 136(2):248–257, 2010.

D. I. C. Covas, H. M. Ramos, N. Graham, and Čedo Maksimović. Application of hydraulic transients for leak detection in water supply systems. *Water Science and Technology: Water Supply*, 4(5):365–374, 2004.

D. I. C. Covas, H. M. Ramos, and A. B. de Almeida. Standing Wave Difference Method for Leak Detection in Pipeline Systems. *Journal of Hydraulic Engineering*, 131(12):1106–1116, 2005.

M. À. Cugueró-Escofet, V. Puig, J. Quevedo, and J. Blesa. Optimal Pressure Sensor Placement for Leak Localisation Using a Relaxed Isolation Index : Application to the Barcelona Water Network. *9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'15)*, 00:1–10, 2015a.

M. À. Cugueró-Escofet, D. García, J. Quevedo, V. Puig, S. Espin, and J. Roquet. A methodology and a software tool for sensor data validation/reconstruction: Application to the catalonia regional water network. *Control Engineering Practice*, 49:159–172, 2016.

P. Cugueró-Escofet, J. Blesa, R. Pérez, M. À. Cugueró-Escofet, and G. Sanz. Assessment of a Leak Localization Algorithm in Water Networks under Demand Uncertainty. *IFAC-PapersOnLine*, 48(21):226–231, 2015b.

W. W. Daniel et al. *Applied nonparametric statistics*. Houghton Mifflin, 1978.

A. Daoudi, M. Benbrahim, and K. Benjelloun. An Intelligent System to Classify Leaks in Water Distribution Pipes. *Preceedings of World Academy of Science, Engineering and Technology*, 4(February):4–6, 2005.

D. De Silva, J. Mashford, and S. Burn. Computer Aided Leak Location and Sizing in Pipe Networks. *Urban Water Security Research Alliance Technical*, 2011(17), 2011.

K. Deep, K. P. Singh, M. L. Kansal, and C. Mohan. A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation*, 212(2):505–518, 2009.

J. A. Delgado-Aguiñaga, O. Begovich, and G. Besançon. Exact-differentiation-based leak detection and isolation in a plastic pipeline under temperature variations. *Journal of Process Control*, 42:114–124, 2016.

R. Díaz-Uriarte and S. A. De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

D. G. Eliades and M. M. Polycarpou. Leakage fault detection in district metered areas of water distribution systems. *Journal of Hydroinformatics*, 14(4):992, 2012.

A. C. D. Escalera, L. E. Garza-Castañón, and A. Vargas-Martínez. Multi-leak detection with wavelet analysis in water distribution networks. In *20th Mediterranean Conference on Control & Automation (MED)*, pages 1155–1160. IEEE, 2012.

M. Fahmy and O. Moselhi. Automated Detection and Location of Leaks in Water Mains Using Infrared Photography. *Journal of Performance of Constructed Facilities*, 24(3):242–248, 2010.

P. V. Fanner, R. Sturm, J. Thornton, R. Liemberger, S. E. Davis, and T. Hoogerwerf. *Leakage Management Technologies.* Water Research Foundation Report Series. IWA Publishing, 2007.

B. Farley, S. R. Mounce, and J. B. Boxall. Field testing of an optimal sensor placement methodology for event detection in an urban water distribution network. *Urban Water Journal*, 7(6):345–356, 2010.

B. Farley, S. R. Mounce, and J. B. Boxall. Development and Field Validation of

a Burst Localization Methodology. *Journal of Water Resources Planning and Management*, 139(December):604–613, 2013.

M. Farley and S. Hamilton. Non-intrusive leak detection in large diameter, low-pressure non-metallic pipes: are we close to finding the perfect solution? *Proceedings of the IWA World Water ...*, pages 1–9, 2008.

M. Ferrante and B. Brunone. Pipe system diagnosis and leak detection by unsteady-state tests. 1. harmonic analysis. *Advances in Water Resources*, 26(1):95–105, 2003a.

M. Ferrante and B. Brunone. Pipe system diagnosis and leak detection by unsteady-state tests. 2. wavelet analysis. *Advances in Water Resources*, 26(1):107–116, 2003b.

M. Ferrante, B. Brunone, and S. Meniconi. Wavelets for the analysis of transient pressure signals for leak detection. *Journal of hydraulic engineering*, 133(11): 1274–1282, 2007.

M. Ferrante, B. Brunone, S. Meniconi, B. W. Karney, and C. Massari. Leak Size, Detectability and Test Conditions in Pressurized Pipe Systems. *Water Resources Management*, pages 4583–4598, 2014.

F. Fusco and A. Ba. Fault diagnosis of water distribution networks based on state-estimation and hypothesis testing. *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 886–892, 2012.

Y. Gao, M. J. Brennan, and P. F. Joseph. On the effects of reflections on time delay estimation for leak detection in buried plastic water pipes. *Journal of Sound and Vibration*, 325(3):649–663, 2009.

J. Gertler, J. Romera, V. Puig, and J. Quevedo. Leak detection and isolation in water distribution networks using principal component analysis and structured residuals. *2010 Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 191–196, 2010.

C. M. Giorgio Bort, M. Righetti, and P. Bertola. Methodology for leakage isolation using pressure sensitivity and correlation analysis in water distribution systems. *Procedia Engineering*, 89:1561–1568, 2014.

Global Water Intelligence. Water tariff survey 2008. Technical report, Oxford, 2008.

J. A. Goulet, S. Coutu, and I. F. C. Smith. Model falsification diagnosis and sensor placement for leak detection in pressurized pipe networks. *Advanced Engineering Informatics*, 27(2):261–269, 2013.

B. Greyvenstein and J. E. van Zyl. An experimental investigation into the pressure-leakage relationship of some failed water pipes. *Journal of Water Supply: Research and Technology-AQUA*, 56(2):117–124, 2007.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

M. Hollander, D. A. Wolfe, and E. Chicken. *Nonparametric statistical methods*. John Wiley & Sons, 2013.

Z. Hu, Y. Bao, T. Xiong, and R. Chiong. Hybrid filter-wrapper feature selection for short-term load forecasting. *Engineering Applications of Artificial Intelligence*, 40:17 – 27, 2015. ISSN 0952-1976.

O. Hunaidi and P. Giamou. Ground-penetrating radar for detection of leaks in buried plastic water distribution pipes. In *Seventh International Conference on Ground-Penetrating Radar, Lawrence, Kansas*, pages 783–786, 1998.

O. Hunaidi, W. Chu, A. Wang, and W. Guan. Detecting leaks in plastic pipes. *Journal / American Water Works Association*, 92(2):82–94, 2000.

C. Hutton and Z. Kapelan. Real-time burst detection in Water Distribution Systems using a Bayesian demand forecasting methodology. *Procedia Engineering*, 119(1): 13–18, 2015.

H. H. Inbarani, A. T. Azar, and G. Jothi. Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis. *Computer methods and programs in biomedicine*, 113(1):175–185, 2014.

R. Ionel, A. Ignea, and S. Ionel. Remote automatic selection of suitable frequency intervals for improved leak detection. *9th RoEduNet IEEE International Conference*, pages 246–251, 2010.

Y. Ishido and S. Takahashi. A new indicator for real-time leak detection in water distribution networks: Design and simulation validation. *Procedia Engineering*, 89:411–417, 2014.

M. S. Islam, R. Sadiq, M. J. Rodriguez, A. Francisque, H. Najjaran, and M. Hoorfar. Leakage detection and location in water distribution systems using a fuzzy-based methodology. *Urban Water Journal*, 8(6):351–365, 2011.

T. N. Jensen and C. S. Kallesøe. Application of a novel leakage detection framework for municipal water supply on aau water supply lab. In *2016 3rd Conference onControl and Fault-Tolerant Systems (SysTol)*. IEEE, 2016.

D. Jung and K. Lansey. Water distribution system burst detection using a nonlinear Kalman filter. *Journal of Water Resources Planning and Management*, on line pu (10.1061/(ASCE)WR.1943-5452.0000464 , 04014070):1–13, 2014.

Y. A. Khulief, A. Khalifa, R. Ben Mansour, and M. A. Habib. Acoustic Detection of Leaks in Water Pipelines Using Measurements inside Pipe. *Journal of Pipeline Systems Engineering and Practice*, 3(2):47–54, 2012.

Y. Kim, S. J. Lee, T. Park, G. Lee, J. C. Suh, and J. M. Lee. Robust leakage detection and interval estimation of location in water distribution network. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 48(8):1264–1269, 2015.

B. Kingdom, R. Liemberger, and P. Marin. The Challenge of Reducing Non-Revenue Water (NRW) in Developing Countries How the Private Sector Can Help: A Look

at Performance-Based Service Countries. *Water supply and sanitation sector board discussion paper series*, 8, 2006.

J. P. C. Kleijnen. Regression and kriging metamodels with their experimental designs in simulation: A review. *European Journal of Operational Research*, 256(1):1 – 16, 2017. ISSN 0377-2217.

A. Lambert. Assessing non-revenue water and its practical component: a practical approach. *Water 21*, June(2), 2003.

D. Laucelli, M. Romano, D. A. Savić, and O. Giustolisi. Detecting anomalies in water distribution networks using EPR modelling paradigm. *Journal of Hydroinformatics*, 18(3):409–427, 2016.

P. J. Lee, J. P. Vítkovský, M. F. Lambert, A. R. Simpson, and J. A. Liggett. Leak location in pipelines using the impulse response function. *Journal of Hydraulic Research*, 45(5):643–652, 2007.

S. J. Lee, G. Lee, J. C. Suh, and J. M. Lee. Online Burst Detection and Location of Water Distribution Systems and Its Practical Applications. *Journal of Water Resources Planning and Management*, 142(1):1–11, 2016.

R. Li, H. Huang, K. Xin, and T. Tao. A Review of Methods for Burst/Leakage Detection and Location in Water Distribution Systems. *Water Science & Technology: Water Supply*, 15(3):429–441, 2015.

X. Li and G. Li. Leak detection of municipal water supply network based on the cluster-analysis and fuzzy pattern recognition. In *E-Product E-Service and E-Entertainment (ICEEE), 2010 International Conference on*, pages 1–5. IEEE, 2010.

J. A. Liggett and L. Chen. Inverse transient analysis in pipe networks. *Journal of Hydraulic Engineering*, 120(8):934–955, 1994.

T. Liu, H. Wei, K. Zhang, and W. Guo. Mutual information based feature selection

for multivariate time series forecasting. In *Control Conference (CCC), 2016 35th Chinese*, pages 7110–7114. IEEE, 2016.

S. N. Lophaven, H. B. Nielsen, and J. Søndergaard. Dace-a matlab kriging toolbox, version 2.0. Technical report, Technical University of Denmark, 2002.

A. Martini, M. Troncossi, and A. Rivola. Automatic Leak Detection in Buried Plastic Pipes of Water Supply Networks by Means of Vibration Measurements. *Shock and Vibration*, 2015:11–15, 2015.

J. Mashford, D. De Silva, S. Burn, and D. Marney. Leak Detection in Simulated Water Pipe Networks Using SVM. *Applied Artificial Intelligence*, 26(5):429–444, 2012.

MATLAB. *MATLAB R2015a version*. The MathWorks Inc., Natick, Massachusetts, 2015.

G. Mazzolani, L. Berardi, D. Laucelli, A. Simone, R. Martino, and O. Giustolisi. Estimating Leakages in Water Distribution Networks Based Only on Inlet Flow Data. *Journal of Water Resources Planning and Management*, pages 1–11, 2015.

B. Mergelas and G. Henrich. Leak locating method for precommissioned transmission pipelines: North American case studies. *Leakage 2005*, 2005.

J. Meseguer, J. M. Mirats, G. Cembrano, V. Puig, and E. Bonada. Leakage detection and localization method based on a hybrid inverse/direct modelling approach suitable for handling multiple-leak scenarios. In *International Conference of Hydroinformatics*. CUNY Academic Works, 2014.

J. Meseguer, J. M. Mirats-Tur, G. Cembrano, and V. Puig. Model-based monitoring techniques for leakage localization in distribution water networks. *Procedia Engineering*, 119(1):1399–1408, 2015.

D. Misiunas, J. Vítkovský, G. Olsson, A. Simpson, and M. Lambert. Burst detection and location in pipe networks using a continuous monitoring technique. *International Conference on Pressure Surges*, pages 24–26, 2004.

D. Misiunas, J. Vítkovský, G. Olsson, M. Lambert, and A. Simpson. Failure monitoring in water distribution networks. *Water science and technology*, 53(4-5): 503–511, 2006.

W. Moczulski, R. Wyczółkowski, K. Ciupke, P. Przystałka, P. Tomasik, and D. Wachla. A methodology of leakage detection and location in water distribution networks – the case study. In *2016 3rd Conference onControl and Fault-Tolerant Systems (SysTol)*, pages 325–330. IEEE, 2016.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2012.

S. R. Mounce, A. Khan, A. S. Wood, A. J. Day, P. D. Widdop, and J. Machell. Sensor-fusion of hydraulic data for burst detection and location in a treated water distribution system. *Information Fusion*, 4(3):217–229, 2003.

S. R. Mounce, J. B. Boxall, and J. Machell. Development and Verification of an Online Artificial Intelligence System for Detection of Bursts and Other Abnormal Flows. *Journal of Water Resources Planning and Management*, 136(3):309–318, 2010.

S. R. Mounce, R. B. Mounce, and J. B. Boxall. Novelty detection for time series data analysis in water distribution systems using support vector machines. *Journal of Hydroinformatics*, 13:672, 2011.

S. R. Mounce, R. B. Mounce, T. Jackson, J. Austin, and J. B. Boxall. Pattern matching and associative artificial neural networks for water distribution system time series data analysis. *Journal of Hydroinformatics*, 16(3):617–632, 2014.

S. R. Mounce, C. Pedraza, T. Jackson, P. Linford, and J. B. Boxall. Cloud based machine learning approaches for leakage assessment and management in smart water networks. *Procedia Engineering*, 119(1):43–52, 2015.

W. Mpesha, S. L. Gassman, and M. H. Chaudhry. Leak detection in Pipes by

Frequency Response Method. *Journal of Hydraulic Engineering*, 127(February): 134–147, 2001.

H. E. Mutikanga, S. K. Sharma, and K. Vairavamoorthy. Review of Methods and Tools for Managing Losses in Water Distribution Systems. *Journal of Water Resources Planning and Management*, 139(April):166–174, 2012.

I. Narayanan, A. Vasan, V. Sarangan, J. Kadengal, and A. Sivasubramaniam. Little Knowledge Isn't Always Dangerous – Understanding Water Distribution Networks Using Centrality Metrics. *IEEE Transactions on Emerging Topics in Computing*, 6750(2):1–1, 2014a.

I. Narayanan, A. Vasan, V. Sarangan, and A. Sivasubramaniam. One meter to find them all: water network leak localization using a single flow meter. *Proc. IPSN*, pages 47–58, 2014b.

A. Nasir, B. H. Soong, and S. Ramachandran. Framework of WSN based human centric cyber physical in-pipe water monitoring system. *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010*, pages 1257–1261, 2010.

S. T. N. Nguyen, J. Gong, M. F. Lambert, A. C. Zecchin, and A. R. Simpson. Least squares deconvolution for leak detection with a pseudo random binary sequence excitation. *Mechanical Systems and Signal Processing*, 99:846–858, 2018.

W. Nixon, M. S. Ghidaoui, and A. A. Kolyshkin. Range of validity of the transient damping leakage detection method. *Journal of Hydraulic Engineering*, 132(9): 944–957, 2006.

A. Nowicki and M. Grochowski. Kernel PCA in application to leakage detection in drinking water distribution system. *Proceedings of the Third international conference on Computational collective intelligence: technologies and applications - Volume Part I*, pages 497–506, 2011.

I. S. Oh, J. S. Lee, and B. R. Moon. Hybrid genetic algorithms for feature selection. *IEEE transactions on pattern analysis and machine intelligence*, 26(11): 1424–1437, 2004.

S. Oreski and G. Oreski. Genetic algorithm-based heuristic for feature selection in credit risk assessment. *Expert systems with applications*, 41(4):2052–2064, 2014.

A. Ostfeld, E. Salomons, L. Ormsbee, J. G. Uber, C. M. Bros, P. Kalungi, R. Burd, B. Zazula-Coetzee, T. Belrain, D. Kang, K. Lansey, H. Shen, E. McBean, Z. Y. Wu, T. Walski, S. Alvisi, M. Franchini, J. P. Johnson, S. R. Ghimire, B. D. Barkdoll, T. Koppel, A. Vassiljev, J.H. Kim, G. Chung, D. G. Yoo, K. Diao, Y. Zhou, J. Li, Z. Liu, K. Chang, J. Gao, S. Qu, Y. Yuan, T. D. Prasad, D. Laucelli, L. S. V. Lyroudia, Z. Kapelan, D. A. Savić, L. Berardi, G. Barbaro, O. Giustolisi, M. Asadzadeh, B. A. Tolson, and R. McKillop. Battle of the Water Calibration Networks (BWCN). *Journal of water resources planning and management*, 138 (October):523–532, 2012.

C. V. Palau, F. J. Arregui, and M. Carlos. Burst Detection in Water Networks Using Principal Component Analysis. *Journal of Water Resources Planning and Management*, 138(February):47–54, 2012.

H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

L. S. Perelman, W. Abbas, X. Koutsoukos, and S. Amin. Sensor placement for fault location identification in water networks: A minimum test cover approach. *Automatica*, 72:166–176, 2016.

R. Pérez and G. Sanz. Modelling and simulation of drinking-water networks. In *Real-time Monitoring and Operational Control of Drinking-Water Systems*, pages 37–52. Springer, 2017.

R. Pérez, V. Puig, J. Pascual, J. Quevedo, E. Landeros, and A. Peralta. Leakage isolation using pressure sensitivity analysis in water distribution networks: Ap-

plication to the barcelona case study. In *12th IFAC Symposium on Large-Scale Systems: Theory and Applications*. International Federation of Automatic Control, 2010.

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

R. Pérez, M. À. Cugueró-Escofet, J. Cugueró-Escofet, and G. Sanz. Accuracy assessment of leak localisation method depending on available measurements. *Procedia Engineering*, 70:1304–1313, 2014.

R. Pérez, J. Cugueró-Escofet, J. Blesa, M. À. Cugueró-Escofet, and G. Sanz. Uncertainty effect on leak localisation in a dma. In *2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 319–324. IEEE, 2016.

Z. Poulakis, D. Valougeorgis, and C. Papadimitriou. Leakage detection in water pipe networks using a Bayesian probabilistic framework. *Probabilistic Engineering Mechanics*, 18:315–327, 2003.

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

P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125, 1994.

R. Puust, Z. S. Kapelan, D. A. Savić, and T. Koppel. Probabilistic Leak Detection in Pipe Networks Using the SCEM-UA Algorithm. *8th Annual Water Distribution Systems Analysis Symposium*, pages 1–12, 2006.

R. Puust, Z. Kapelan, D. A. Savić, and T. Koppel. A review of methods for leakage management in pipe networks. *Urban Water Journal*, 7(March 2015):25–45, 2010.

J. Quevedo, M. À. Cugeró-Escofet, R. Pérez, F. Nejjari, V. Puig, and J. M. Mirats. Leakage location in water distribution networks based on correlation measurement

of pressure sensors. *8th IWA Symposium on Systems Analysis and Integrated Assessment*, 2012.

M. Quiñero Grueiro, C. Verde, and A. Prieto-Moreno. Leaks' detection in water distribution networks with demand patterns. In *2016 3rd Conference onControl and Fault-Tolerant Systems (SysTol)*, pages 307–312. IEEE, 2016.

A. Rajeswaran, S. Narasimhan, and S. Narasimhan. A graph partitioning algorithm for leak detection in water distribution networks. *Computers & Chemical Engineering*, 2017.

I. Rojek and J. Studziński. Comparision of different types of neuronal nets for failures locaion within water-supply networks. *Eksploatacja i Niezawodność*, 16 (1):42–47, 2014.

J. Roma, R. Pérez, G. Sanz, and S. Grau. Model calibration and leakage assessment applied to a real Water Distribution Network. *Procedia Engineering*, 119(1):603–612, 2015.

M. Romano, Z. Kapelan, and D. A. Savić. Automated detection of pipe bursts and other events in water distribution systems. *Journal of Water Resources Planning and Management*, 140(4):457–467, 2012.

M. Romano, Z. Kapelan, and D. A. Savić. Geostatistical techniques for approximate location of pipe burst events in water distribution systems. *Journal of Hydroinformatics*, 15(3):634–651, 2013a.

M. Romano, Z. Kapelan, and D. A. Savić. Evolutionary algorithm and expectation maximization strategies for improved detection of pipe bursts and other events in water distribution systems. *Journal of Water Resources Planning and Management*, 140(5):572–584, 2013b.

M. Romano, K. Woodward, and Z. Kapelan. Statistical Process Control Based System for Approximate Location of Pipe Bursts and Leaks in Water Distribution Systems. *Procedia Engineering*, 186:236–243, 2017.

A. Rosich, V. Puig, and M. V. Casillas. Leak localization in drinking water distribution networks using structured residuals. *International Journal of Adaptive Control and Signal Processing*, 28(21):991–1007, 2014.

L. A. Rossman. EPANET 2: users manual. *Cincinnati US Environmental Protection Agency National Risk Management Research Laboratory*, 38, 2000.

Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

D. Sala and P. Ko. Detection of leaks in a small-scale water distribution network based on pressure data - experimental verification. *Procedia Engineering*, 70: 1460–1469, 2014.

A. Salmerón, A. L. Madsen, F. Jensen, H. Langseth, T. D. Nielsen, D. Ramos-López, A. M. Martínez, and A. Masegosa. Parallel filter-based feature selection based on balanced incomplete block designs. In *Proceedings of the European Conference on Artificial Intelligence (ECAI 2016)*, 2016.

A. Sánchez-Fernández, M. J. Fuente, and G. I. Sainz-Palmero. Fault detection with Distributed PCA methods in Water Distribution Networks. *2015 23th Mediterranean Conference on Control and Automation (MED)*, pages 156–161, 2015.

G. Sanz and R. Pérez. 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(1977):1–9, 2015.

G. Sanz, R. Pérez, and A. Escobet. Leakage localization in water networks using fuzzy logic. *2012 20th Mediterranean Conference on Control & Automation (MED)*, pages 646–651, 2012.

G. Sanz, R. Pérez, Z. Kapelan, and D. A. Savić. Leak Detection and Localization through Demand Components Calibration. *Journal of Water Resources Planning and Management*, 142(2), 2015.

R. Sarrate, F. Nejjari, and A. Rosich. Sensor placement for fault diagnosis performance maximization under budgetary constraints. *International Conference on Systems and Control*, pages 178–183, 2012.

R. Sarrate, J. Blesa, and F. Nejjari. Clustering techniques applied to sensor placement for leak detection and location in water distribution networks. In *2014 22nd Mediterranean Conference of Control and Automation (MED)*, pages 109–114. IEEE, 2014a.

R. Sarrate, J. Blesa, F. Nejjari, and J. Quevedo. Sensor placement for leak detection and location in water distribution networks. *Water Science and Technology: Water Supply*, 14(5):795–803, 2014b.

R. Sarrate, J. Blesa, and F. Nejjari. Sensor Placement for Leak Monitoring in Drinking Water Networks combining Clustering Techniques and a Semi-Exhaustive Search. In *2016 3rd Conference onControl and Fault-Tolerant Systems (SysTol)*, pages 419–424. IEEE, 2016.

D. A. Savić, Z. Kapelan, and P. M. R. Jonkergouw. Quo vadis water distribution model calibration? *Urban Water Journal*, 6(March 2015):3–22, 2009.

B. Shakmak and A. Al-Habaibeh. Detection of water leakage in buried pipes using infrared technology; a comparative study of using high and low resolution infrared cameras for evaluating distant remote detection. In *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–7. IEEE, 2015.

A. K. Soares, D. I. C. Covas, and H. M. Ramos. Damping analysis of hydraulic transients in pump-rising main systems. *Journal of Hydraulic Engineering*, 139 (2):233–243, 2012.

J. Sousa, L. Ribeiro, J. Muranho, and A. Sá Marques. Locating leaks in water distribution networks with simulated annealing and graph theory. *Procedia Engineering*, 119(1):63–71, 2015.

S. Srirangarajan, M. Allen, A. Preis, M. Iqbal, H. B. Lim, and A. J. Whittle. Wavelet-based burst event detection and localization in water distribution systems. *Journal of Signal Processing Systems*, 72(1):1–16, 2013.

A. Stampolidis, P. Soupios, F. Vallianatos, and G. N. Tsokas. Detection of leaks in buried plastic water distribution pipes in urban places - a case study. *Proceedings of the 2nd International Workshop onAdvanced Ground Penetrating Radar, 2003.*, pages 14–16, 2003.

D. B. Steffelbauer and D. Fuchs-Hanusch. Efficient Sensor Placement for Leak Localization Considering Uncertainties. *Water Resources Management*, 30(14):5517–5533, 2016.

M. A. Stephens. Edf statistics for goodness of fit and some comparisons. *Journal of the American statistical Association*, 69(347):730–737, 1974.

J. Tang, S. Alelyani, and H. Liu. Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, page 37, 2014.

T. Tao, H. Huang, F. Li, and K. Xin. Burst Detection Using an Artificial Immune Network in Water-Distribution Systems. *Journal of Water Resources Planning and Management*, 140(10):1–10, 2014.

P. van Thienen. A method for quantitative discrimination in flow pattern evolution of water distribution supply areas with interpretation in terms of demand and leakage. *Journal of Hydroinformatics*, 15(1):86, 2013.

C. Verde. Accommodation of multi-leak location in a pipeline. *Control Engineering Practice*, 13(8):1071–1078, 2005.

J. R. Vergara and P. A. Estévez. A review of feature selection methods based on mutual information. *Neural computing and applications*, 24(1):175–186, 2014.

J. P. Vítkovský, A. R. Simpson, and M. F. Lambert. Leak detection and calibration using transients and genetic algorithms. *Journal of Water Resources Planning and Management*, 126(4):262–265, 2000.

J. P. Vítkovský, J. A. Liggett, A. R. Simpson, and M. F. Lambert. Optimal Measurement Site Locations for Inverse Transient Analysis in Pipe Networks. *Journal of Water Resources Planning and Management*, 129(6):480–492, 2003.

D. Wachla, P. Przystalka, and W. Moczulski. A Method of Leakage Location in Water Distribution Networks using Artificial Neuro-Fuzzy System. *IFAC-PapersOnLine*, 48(21):1216–1223, 2015.

T. M. Walski, D. V. Chase, D. A. Savić, W. Grayman, S. Beckwith, and E. Koelle. *Advanced water distribution modeling and management*. Haestad press, 2003.

X. Wang, M. F. Lambert, A. R. Simpson, J. A. Liggett, and J. P. Vítkovský. Leak Detection in Pipelines using the Damping of Fluid Transients. *Journal of Hydraulic Engineering*, 128(7):697–711, 2002.

Y. Wu and S. Liu. A review of data-driven approaches for burst detection in water distribution systems. *Urban Water Journal*, 9006(March):1–12, 2017.

Z. Y. Wu and P. Sage. Water loss detection via genetic algorithm optimization-based model calibration. *ASCE 8th Annual International*, 5(2005):1–11, 2006.

Z. Y. Wu, M. Farley, D. Turtle, Z. Kapelan, J. Boxall, S. Mounce, S. Dahasahasra, M. Mulay, and Y Kleiner. *Water Loss Reduction*. Bentley Institute Press, 2011.

B. Wysogląd and R. Wyczółkowski. An optimization of heuristic model of water supply. *Computer Assisted Mechanics and Engineering Sciences*, 14(4):767–776, 2007.

D. L. Xu, J. Liu, J. B. Yang, G. P. Liu, J. Wang, I. Jenkinson, and J. Ren. Inference and learning methodology of belief-rule-based expert system for pipeline leak detection. *Expert Systems with Applications*, 32(1):103–113, 2007.

B. Xue, M. Zhang, and W. N. Browne. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, 43(6):1656–1671, 2013.

B. Xue, M. Zhang, W. N. Browne, and X. Yao. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626, 2016.

J. Yang, Y. Wen, and P. Li. Leak location using blind system identification in water distribution pipelines. *Journal of Sound and Vibration*, 310(1):134–148, 2008.

J. Yang, Y. Wen, and P. Li. Information processing for leak detection on underground water supply pipelines. *3rd International Workshop on Advanced Computational Intelligence*, pages 623–629, 2010.

G. Ye and R. A. Fenner. Kalman Filtering of Hydraulic Measurements for Burst Detection in Water Distribution Systems. *Journal of Pipeline Systems Engineering and Practice*, 2(1):14–22, 2011.

G. Ye and R. A. Fenner. Study of Burst Alarming and Data Sampling Frequency in Water Distribution Networks. *Journal of Water Resources Planning and Management*, 140(6):1–7, 2014a.

G. Ye and R. A. Fenner. Weighted Least Squares with Expectation-Maximization Algorithm for Burst Detection in U . K . Water Distribution Systems. *Journal of Water Resources Planning and Management*, 140(April):417–424, 2014b.

L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of machine learning research*, 5(Oct):1205–1224, 2004.

T. T. T. Zan, H. B. Lim, K. Wong, A. J. Whittle, and B. Lee. Event Detection and Localization in Urban Water Distribution Network. *IEEE Sensors Journal*, 14(12):4134–4142, 2014.

H. Zhang and L. Wang. Leak detection in water distribution systems using Bayesian theory and Fisher's law. *Transactions of Tianjin University*, 17:181–186, 2011.

J. L. Zhang and W. X. Guo. Study on the characteristics of the leakage acoustic emission in cast iron pipe by experiment. *Proceedings - 2011 International Confer-*

ence on Instrumentation, Measurement, Computer, Communication and Control, IMCCC 2011, pages 122–125, 2011.

Z. Zhong, J. Suzuki, T. Miyake, H. Kondou, and K. Enastu. Study on water leak detection using wavelet instantaneous cross-correlation. In *2015 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, pages 133–137. IEEE, 2015.

Z. J. Zhou, C. H. Hu, D. L. Xu, J. B. Yang, and D. Zhou. Bayesian reasoning approach based recursive algorithm for online updating belief rule based expert system of pipeline leak detection. *Expert Systems with Applications*, 38(4):3937–3943, 2011.