UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

A methodology for the automation of building intelligent process control systems

Josep Pascual Pañach

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

PhD program in Artificial Intelligence

A Methodology for The Automation of Building Intelligent Process Control Systems

Doctoral thesis by:

Josep Pascual Pañach

Thesis advisors:

Miquel Sànchez Marrè

Miquel Àngel Cugueró Escofet

Computer Science Department

Barcelona, July 2023

Acknowledgments

Arriba el final d’una etapa marcada principalment per la tesi doctoral, i durant la qual he pogut conèixer i treballar amb moltíssimes persones que m’han acompanyat en aquest camí, no sempre fàcil. És per això que vull agrair tot el suport i ajuda que he rebut durant aquests anys de totes aquestes persones. I molt especialment ...

Als meus directors, Miquel Sànchez Marrè i Miquel Àngel Cugueró, per la vostra paciència i haver-me guiat en aquest camí que s’ha anat allargant i allargant. I per les revisions altes hores de la matinada.

Al Consorci Besòs Tordera i CCB Serveis Mediambientals, per què m’heu obert les portes i ajudat en tot allò que he demanat. Tant a la gent de les plantes depuradores, com al Departament d’Operacions, Informàtica i Innovació, que sempre han buscat algun forat en les seves atrafegades agendes per discutir qualsevol cosa relacionada amb aquest treball.

Als amics i companys de les classes de cuina, per les bones estones de desconnexió i bons sopars que cada setmana compartim.

A la mare i familiars, pel seu suport incondicional durant tots aquests anys, però sobretot durant les últimes setmanes, que han estat força dures. I molt especialment a la meva parella Cristina, que m’ha animat en els moments més difícils i sense això no hagués pogut arribar al final, i al nostre petit Martí, que tantes bones estones ens està fent passar.

Moltes gràcies a tots i totes!

This thesis was financially supported by ACCIONA Agua S.A, by the Industrial Doctorate Programme (ref. 2017-DI-048) of the Catalan Agency of University and Research Grants Management (AGAUR), and by the Research Consolidated Groups/Centers Grant from AGAUR, grant numbers 2017 SGR 574 and 2017 SGR 482. The author would also like to acknowledge the support received from CCB Serveis Mediambientals S.A.U and Consorsi Besòs Tordera during the development of this work

Abstract

One of the major problems to design and implement control and supervision systems for real-world processes lies in the need to stablish an ad-hoc solution for each process, i.e., environmental systems in this thesis, more precisely Waste Water Treatment Plants (WWTPs). Despite WWTPs have similarities to some extent, they have particular characteristics that must be considered in the process control system design. This fact involves considerable amounts of time and resources from design stages to maintenance, including implementation and start-up stages.

This thesis proposes an Intelligent Decision Support (IDS) methodology based on the interoperation of a data-driven technique –Case Based Reasoning (CBR)– and a model-driven technique –Rule Based Reasoning (RBR)– for control, supervision and decision support of real-world processes. Design stage of control and decision support tools tends to be somehow ad-hoc regarding to the nature of the processes involved. Hence, an automated approach is proposed for the sake of scalability to different types and configurations. The proposed hybrid scheme provides complementarity in the set-point generation for the process controllers, increasing the reliability of the Intelligent Process Control System (IPCS), which is the core component of the IDSS. The IDSS is flexible to allow coping with the evolution of dynamic systems, learning from recorded situations, through a new temporal reasoning approach. Furthermore, some innovative data imputation techniques are proposed for managing the missing values appearing in a complex process. The approach presented has been implemented in different real facilities in the Barcelona area (Catalonia, Spain).

Resum

Un dels principals problemes alhora de dissenyar i implementar un sistema de supervisió i control per a un procés radica en la necessitat d'establir una solució ad-hoc per a cada instal·lació – Estacions Depuradores d'Aigües Residuals (EDAR) en el context d'aquesta tesi. Malgrat les similituds entre totes les plantes, hi ha característiques particulars de cada sistema que cal tenir en compte en el procés de disseny del sistema de control. Aquest fet implica importants quantitats de temps i recursos destinats tant al disseny com a la implementació, posada en marxa i manteniment de la solució proposada.

Aquesta tesi proposa una metodologia per generar un sistema intel·ligent de suport a la decisió basat en la interoperació de tècniques basades en dades –Raonament Basat en Casos (CBR)– i en models –Raonament Basat en Regles (RBR)– per al control, supervisió i suport a la decisió de sistemes mediambientals. El disseny d'eines per al control i la supervisió sol implicar solucions fetes a mida en funció de la naturalesa dels processos considerats en cada cas. Per a això, es proposa una solució que permet automatitzar aquest disseny i escalar-lo a diferents tipus de processos i configuracions. Així, es proposa una solució híbrida que proporciona redundància en la generació de consignes per al procés, augmentant així la fiabilitat del Sistema Intel·ligent de Control (IPCS, en angles), nucli del Sistema Intel·ligent de Suport a la Decisió (IDSS, en anglès). Addicionalment, es tracta d'una solució flexible capaç d'aprendre de les experiències enregistrades, mitjançant una aproximació de raonament temporal. A més a més, es proposen algunes tècniques innovadores d'imputació de dades mancants que apareixen en un procés complex. La metodologia es materialitza en una eina instal·lada i testejada en diferents instal·lacions reals ubicades a la zona de Barcelona.

List of acronyms

AEAS	Asociación Española de Abastecimientos y Saneamiento
AI	Artificial Intelligence
AMB	Àrea Metropolitana de Barcelona
ANFIS	Adaptative Neuro Fuzzy Inference Systems
ANN	Artificial Neural Network
AR	Association Rules
ARX	Autoregressive with Extra input
ASM	Activated Sludge Model
BN	Bayesian Network
BSM	Benchmark Simulation Model
BOD	Biological Oxygen Demand
CB	Case Base
CBR	Case-Based Reasoning
CBT	Consorci Besòs Tordera
CCBR	Ceaseless Case-Based Reasoning
COD	Chemical Oxygen Demand
CR	Classification Rules
CSV	Comma-Separated Values
DIBA	Diputació de Barcelona
DMG	Data Mining Group
DO	Dissolved Oxygen
DSS	Decision Support System
DT	Decision Tree
DTD	Document Type Definition
DTW	Dynamic Time Warping
DWTP	Drinking Water Treatment Plant
EB	Episodes Base
ED	Euclidean Distance
EDAR	Estació Depuradora d'Aigües Residuals
EDSS	Environmental Decision Support System
EEC	European Economic Community
ELM	Extreme Learning Machines
GA	Genetic Algorithms
GIS	Geographical Information System
GUI	Graphical User Interface
HCBR	Historical Case-Based Reasoning

HTML	HyperText Markup Language
ICT	Information and Communication Technology
IDS	Intelligent Decision Support
IDSS	Intelligent Decision Support System
IEDSS	Intelligent Environmental Decision Support System
IPCS	Intelligent Process Control System
IWA	International Water Association
KB	Knowledge Base
KBR	Knowledge-Based Reasoning
KPI	Key Performance Indicator
LR	Linear Regression
LSTM	Long Short-Term Memory
MICE	Multiple Imputation by Chained Equations
MJC	Minimum Jump Costs
MLR	Multiple Linear Regression
MP	Meta Predictor
NARX	Non-Linear autoregressive with External input
NH_4	Ammonium
NN	Neural Network
NO_3	Nitrate
OPC	OLE for Process Control
OLE	Object Linking and Embedding
PDB	Process Database
PE	Population Equivalent
PID	Proportional Integral Derivative
PLC	Programable Logic Controller
PMML	Predictive Model Markup Language
RBR	Rule Based Reasoning
RD	Royal Decree
RDI	Research, Development and Innovation
RKNN	Reliable K Nearest Neighbour
RMSE	Root Mean Square Error
SCADA	Supervisory Control and Data Acquisition
SES	Socio-Environmental Systems
SI	Special Issue
SS	Suspended Solids
SVR	Support Vector Regression

TCBR	Temporal Case Based Reasoning
WV	Weighted Voting
WWTP	Waste Water Treatment Plant
XML	Extensible Markup Language
XSD	XML Schema Document

List of figures

Figure 2.1 Example of hierarchical memory organization 12
Figure 2.2 Case-Based Reasoning cycle scheme 13
Figure 2.3 Architecture of a Rule-Based Reasoning System including main components 18
Figure 2.4 Integral urban water cycle 23
Figure 2.5 Aerial view of a Waste Water Treatment Plant 24
Figure 2.6 Industry 4.0 diagram. Source: i2CAT (https://i2cat.net) 25
Figure 2.7 WWTP generic scheme 27
Figure 3.1 Architecture of the proposed Intelligent Decision Support Methodology 32
Figure 3.2 Intelligent Process Control structure 33
Figure 3.3 Data science/Model layer: a) data-driven, b) model-driven 35
Figure 3.4 Process design layer 35
Figure 3.5 Process control layer 36
Figure 4.1 Interoperation of RBR, CBR and Decision modules scheme 40
Figure 4.2 Number of retained cases over time. 47
Figure 5.1 Proposed reasoning scheme extending the classic CBR cycle 54
Figure 5.2 CBR vs TCBR problem approaches 58
Figure 6.1 Integration of the CBR-based imputation module in the classic CBR cycle 64
Figure 6.2 Data Validation stage 66
Figure 6.3 Example of dismissed samples in Test 1 67
Figure 6.4 Example of a dismissed sample in Test 2 67
Figure 6.5 Example of a dismissed sample in test 3 68
Figure 6.6 Example of a dismissed sample in Test 4 68
Figure 6.7 Imputation using the retrieved case/episode 70
Figure 6.8 Non-linear autoregressive with external (exogenous) input (NARX) Neural Network 72
Figure 6.9 Voting algorithm 73
Figure 6.10 Weighted voting algorithm 74
Figure 6.11 Stacking algorithm 75
Figure 7.1 CBT location and area of action 78
Figure 7.2 CBT managed area, including municipalities limits and WWTPs 79
Figure 7.3 Location of Santa Maria de Palautordera WWTP in the CBT river basin 81
Figure 7.4 Santa Maria de Palautordera WWTP layout 81
Figure 7.5 Location of Castellar WWTP in the CBT river basin 84
Figure 7.6 Castellar WWTP layout 84
Figure 7.7 Data collection architecture 86
Figure 7.8 OPC client window highlighting relevant items read from the PLC 86
Figure 7.9 Real faults during a period of three months 87

Figure 7.10 Identification of invalid values in a raw data set.. 90
Figure 7.11 Main screen of the IDSS (relevant parts).. 92
Figure 7.12 Configuration panels of the RBR and CBR modules in the IDSS (relevant parts)................ 92
Figure 7.13 Solutions panel in the IDSS (relevant parts).. 93
Figure 7.14 CB panel in the IDSS (relevant parts).. 93
Figure 7.15 KPIs panel in the IDSS (relevant parts).. 94
Figure 7.16 CBR performance panel in the IDSS (relevant parts)... 94
Figure 7.17 Case retrieval panel in the IDSS (relevant parts).. 95
Figure 7.18 Implementation of the Process control layer using the Simulink environment 95
Figure 7.19 Example of distance threshold calculation using real data 97
Figure 7.20 Automatic sampler: configuration panel (left) and interior of the sampler with 24 bottles for samples.. 98
Figure 7.21 Ammonium, nitrogen and electrical consumption KPI baseline (before IDSS)..................... 99
Figure 7.22 NH4 24h average (24MA-AC) KPI for the test period.. 100
Figure 7.23 BEC KPI for the test period.. 101
Figure 7.24 Total Nitrogen (TNC) KPI and Nitrogen removal efficiency for the test period................. 102
Figure 7.25 Example of a 6 samples length episode 104
Figure 7.26 Accuracy of the temporal CBR solution (left); Accuracy for ANN and LR models (right). 105
Figure 7.27 Calibration of episode length and feature weights for data imputation using TCBR and comparison with the non-calibrated module. 107
Figure 7.28 Test with unfaulty data considering three weeks from the validation dataset...................... 108
Figure 7.29 Test with faulty data considering three weeks from the validation dataset: a) one hour missing value periods; b) and c) two hour missing value periods 109
Figure 7.30. Comparison of measured and predicted values using different models with unfaulty data. 111
Figure 7.31. Evaluation of different models using faulty data (fault #4 in Table 7.11).......................... 111

List of tables

Table 2.1 NOT operator 20

Table 2.2 AND / OR operators.. 20

Table 7.1 Influent characterization for Santa Maria de Palautordera WWTP... 80

Table 7.2 Available sensors for Santa Maria de Palautordera WWTP 82

Table 7.3 Influent characterization for Castellar del Vallès WWTP... 83

Table 7.4 Available sensors of Castellar del Vallès WWTP.. 85

Table 7.5 Data mining techniques and reasoning methods 88

Table 7.6 Percentage of dismissed samples in time-series from Figure 7.10.. 90

Table 7.7 Competence of the reasoning system 103

Table 7.8 Performance of the TCBR approach: accuracy of the problem resolution for different datasets and comparison with ANN and LR models 105

Table 7.9 Considered variables from ASM1 model.. 107

Table 7.10 RMSE for different faults in Figure 7.29. 109

Table 7.11. Models' performance comparison for different faults in the ammonium sensor data........... 112

Table of contents

Acknowledgments i
Abstract v
Resum vii
List of acronyms ix
List of figures xiii
List of tables xv

1. INTRODUCTION 1

1.1. INDUSTRIAL DOCTORATE PLAN 2
1.2. MOTIVATION AND OBJECTIVES 2
1.3. OUTLINE OF THE THESIS 4
1.4. COMPILATION OF GENERATED PUBLICATIONS 4
1.5. CONTRIBUTIONS 6

2. BACKGROUND AND STATE OF THE ART 7

2.1. INTRODUCTION 8
2.2. INTELLIGENT DECISION SUPPORT SYSTEMS 8
2.3. CASE-BASED REASONING 10
2.3.1 Organization and Representation of cases 11
2.3.2 Case-Based Reasoning Cycle 12
2.3.3 Case Retrieval and Similarity Assessment 13
2.3.4 Case Adaptation 14
2.3.5 Case Evaluation 15
2.3.6 Case Learning 16
2.4. TEMPORAL CASE-BASED REASONING 17
2.5. RULE-BASED REASONING 18
2.5.1 Fact Base 19
2.5.2 Knowledge Base 19
2.5.3 The Inference Engine 21
2.5.4 Meta-Reasoning Component 22
2.6. APPLICATION DOMAIN: WASTE WATER TREATMENT 23
2.6.1 Integral Water Cycle 23
2.6.2 Industry 4.0 in the water sector 25
2.6.3 Waste Water Treatment Plants 27
2.7. CONCLUSIONS 29

3. AN IDSS METHODOLOGY FOR INTELLIGENT PROCESS CONTROL SYSTEMS 31

3.1. PROBLEM DESCRIPTION 32
3.2. SYSTEM ARCHITECTURE 32
3.2.1 Data Science flow 34

3.2.2 Process design flow 35
3.2.3 Process control flow 36
3.3. CONCLUSIONS 36

4. INTEGRATED REASONING SYSTEM 39

4.1. RBR APPROACH 41
4.2. CBR APPROACH 41
4.3. COMBINATION OF RBR AND CBR APPROACHES 48
4.4. CONCLUSIONS 49

5. TEMPORAL CBR APPROACH 51

5.1. TEMPORAL CASE-BASED REASONING TERMINOLOGY 52
5.2. TEMPORAL REASONING APPROACH 53
5.3. CONCLUSIONS 58

6. DATA VALIDATION AND IMPUTATION 61

6.1. INTEGRATION OF THE VALIDATION AND IMPUTATION MODULE 63
6.2. PROBLEM DESCRIPTION 64
6.3. VALIDATION STAGE 66
6.4. IMPUTATION STAGE 69
6.4.1 Imputation models 69
6.4.2 Ensemble of models 72
6.5. CONCLUSIONS 75

7. EXPERIMENTAL RESULTS 77

7.1. CASE STUDY 78
7.1.1 Consorci Besòs Tordera 78
7.1.2 Santa Maria de Palautordera WWTP 80
7.1.3 Castellar WWTP 83
7.1.4 Data collection 85
7.2. IDSS IMPLEMENTATION 87
7.2.1 Development environments research 88
7.2.2 Data acquisition and storage 89
7.2.3 Data science layer 90
7.2.4 Process design layer 91
7.2.5 Process control layer 95
7.3. IDSS OPERATION IN REAL FACILITIES 96
7.4. VALIDATION AND IMPUTATION MODULE 106
7.4.1 Results using synthetic data 106
7.4.2 Results using real data from the case study 110
7.5. CONCLUSIONS 112

8. DISCUSSION 115

8.1. DISCUSSION OF THE IDSS IMPLEMENTATION 116
8.2. DISCUSSION OF THE IDSS OPERATION IN REAL FACILITIES 117
8.3. DISCUSSION OF THE VALIDATION AND IMPUTATION MODULE 119

9. CONCLUDING REMARKS 121

9.1. CONCLUSIONS 122
9.2. FUTURE WORK 123

REFERENCES 125

REFERENCES 126

1. Introduction

1.1. Industrial Doctorate Plan

The objectives of the Industrial Doctorate Plan are to contribute to the competitiveness and the internationalization of the Catalan industry, talent recruitment and to place future PhD holders in the right position in the company to carry out Research, Development and Innovation (R&D) projects.

The essential element of the industrial doctorate process is a research project carried out in a company in collaboration with a university or research centre. The Industrial Doctorate act as a bridge for knowledge transfer and encourage closer ties between Catalan industry, universities and research centres by means of a research project in line with the interests of the company. This thesis has been funded by the Catalan Agency of University and Research Grants Management (AGAUR), from Catalan Government.

The more practical nature of an Industrial Doctorate Program than a traditional academic PhD must be noted in the research presented in this thesis. Notwithstanding, the contributions of this thesis are general enough for any complex real-world system. The main contributions of the research done in the framework of this Industrial Doctorate have been implemented in real facilities, Waste Water Treatment Plants (WWTPs) in the case of this work.

To bring research to life is a great opportunity, and also an interesting challenge. On the one hand, the implementation of methods in real applications is appealing. For example, the availability of data from real processes or its quality are not always guaranteed. On the other hand, solutions that will be used in day-to-day operations in the company should consider some specifications from the end users that would not happen in a non-applied research context. In this sense, developed tools or methods should be able to be integrated in the existing operating environment. Regarding the kind of solution and in the particular case of this thesis, the use of explainable Artificial Intelligence (AI) methods is also appreciated by the end users.

1.2. Motivation and objectives

The supervision, control and optimization of processes can be a complex task because of its variability and potential complexity. In a process there may be many elements of different types that interact with each other —e.g., mechanical, electronic, human, biological, chemical, etc.—, sometimes with unknown dynamics, that may turn the control and supervision tasks very complex. In the case of WWTPs, the risk of malfunction is even more severe, since the consequences can be dangerous for the environment and for humans. The quality of the effluent of the plant must comply with different applicable environmental regulations, whether European, regional or local. WWTPs are in operation 24 hours a day, 365 days a year, so the need of using adequate monitoring and intelligent control techniques is essential to ensure a reliable management of the process, improving the diagnosis of possible problems and providing appropriate solutions. Another implication of this continuous operation is that the energy consumption is remarkably high. The generation of set-points to control the process in an efficient way is also important to ensure the sustainability of the waste water treatment process.

One common and important problem when designing a new control / supervision tool for complex dynamic systems is the *ad-hoc* nature of this design for each particular installation, depending on its particular specifications: sub-processes involved, the particular design of each installation and available sensors and actuators. This entails huge amounts of time and resources from design stages to maintenance, including implementation and start-up phases. Artificial Intelligence (AI) methods can provide important improvements to the supervision and control of processes, such as qualitative information management, expert knowledge modelling, uncertainty modelling and reasoning and learning abilities.

This PhD thesis aims to propose a framework to design control and supervision systems for real-world systems allowing an effective deployment in any installation. Hence, the main objectives of this thesis are:

1. The definition of a generic framework for the design and deployment of intelligent control and supervision systems for real world systems (Chapter 3), providing a useful and systematic approach to interoperate different models at different stages of the design, as well as to automate the building of the IDSS.
2. To develop a reasoning system to provide control set-points and diagnosis of the process combining a model-based technique - Rule-Based Reasoning (RBR) here – and a data-driven technique – Case-Based Reasoning (CBR) here (Chapters 4 and 5).
3. To develop a methodology to deal with incorrect or missing data (Chapter 6).
4. To provide methods to evaluate the performance of both the process and the reasoning system, with particular attention in the interaction with the user (Chapter 7).
5. To deploy a prototype application to test the methods in different real facilities, here WWTPs (Chapter 7)

It is also important to highligth that the framework and all the methods proposed in this thesis are generic enough to allow the scalability to any WWTP installation, as well as further types of systems beyond the environmental domain .

Finally, it is necessary to point out that :

- All the data used is recorded from real sources, i.e., data are not obtained from simulation or test environments. Thus, it implies an additional challenge due to different facts, namely: data quality, e.g. noisy or uncalibrated sensors and breakdowns, maintenance tasks. Also, inherent characteristics of working with real installations should be noted, as uncontrolled or exceptional situations such as, in the waste water domain, extreme weather conditions or contaminant discharges over allowed limits.
- Automatic control is sometimes interfered with manual control decided by the plant operators. These actions may be due to guarantee the effluent quality or even driven by the subjective vision of the operator regarding how the process control should perform. They are usually more conservative than the ones proposed by the IDSS.

- Missing information occur when a particular situation of the process cannot be explained or detected with available sensors. Then, the IDSS cannot deal with them and manual control is requiered.

All these situations regarding data availability / quality and boundary conditions can be bypassed or controlled when working on a simulation environment or a pilot scale plant, but must be handled when working with a real scale system.

1.3. Outline of the thesis

The present document, after this introductory chapter, is organised in four main chapters. In Chapter 2, a necessary background and state of the art review about the main topics addressed in this thesis is introduced, namely: Intelligent Decision Support Systems and two AI techniques that are the core of this work – CBR and RBR. The case study is also described, starting from a macro point of view, i.e., Integral Water Cycle, to the last stage of this cycle, the waste water treatment, in which the developed methods are applied. In Chapter 3, a description of the proposed IDSS architecture is presented. In Chapter 4, the reasoning system is described, as well as how Rule-Based Reasoning (RBR) and Case-Based Reasoning (CBR) techniques are combined to solve the control set-points generation. In Chapter 5, a temporal CBR approach is presented. Chapter 6 introduces data validation and imputation methods, a previous and essential step before applying the data-driven methods presented in the preceding sections. Some of the methodologies on which the proposal in Chapter 6 are based, are the ones developed in Chapters 4 and 5. In Chapter 7 the experimental results are presented and discussed. Finally, Chapter 8 summarizes the resulting conclusions derived from this thesis, as well as the future work lines.

1.4. Compilation of generated publications

The list of generated publications during the development of this thesis is given below, classified depending on its status: *published*, *under revision* and *in preparation*.

Published:

[1] Pascual-Pañach, Josep, Cugueró-Escofet, Miquel Àngel, Sànchez-Marrè, Miquel, Interoperating data-driven and model-driven techniques for the automated development of intelligent environmental decision support systems, *Environmental Modelling & Software, Volume 140, 2021, 105021, ISSN 1364-8152, https://doi.org/10.1016/j.envsoft.2021.105021.*

[2] Pascual-Pañach, J., Cugueró-Escofet, M.A., Aguiló-Martos, P., Sànchez-Marrè, M. (2018). An Interoperable Workflow-Based Framework for the Automation of Building Intelligent Process Control Systems. *9th International Congress on Environmental Modelling and Software, Fort Collins, Colorado, USA.*

[3] Pascual-Pañach, J., Cugueró-Escofet, M.A., Sànchez-Marrè, M., Aguiló-Martos, P. (2019). Application of CBR for intelligent process control of a WWTP. *IOS Press, Frontiers in artificial intelligence and applications, 2019, 319, 160 - 169, 0922-6389*

[4] Pascual-Pañach, J., Sànchez-Marrè, M., Cugueró-Escofet, M.A., (2022). Ensemble model-based method for time series sensors' data validation and imputation applied to a real Waste Water Treatment Plant. *11th International Congress on Environmental Modelling and Software, Brussels, Belgium.*

[5] Pascual-Pañach, J., Sànchez-Marrè, M., Cugueró-Escofet, M.A., (2022). Optimizing Online Time-Series Data Imputation Through Case-Based Reasoning. *IOS Press, Frontiers in artificial intelligence and applications, 2022, 356, 87 – 90, 10.3233/FAIA220320*

[6] Josep Pascual-Pañach, Miquel Àngel Cugueró-Escofet, Pere Aguiló-Martos, Miquel Sànchez-Marrè (2018). Herramienta basada en minería de datos para automatización del diseño de sistemas inteligentes en EDAR. *XXXV Congreso AEAS 2019, Valencia, Spain.*

[7] Pascual-Pañach, J., Cugueró-Escofet, M.A., Sànchez-Marrè, M., Aguiló-Martos Martos, P. (2019). Data mining based tool for the automation of the design of intelligent process control systems in waste water treatment plants. *IWA Spain National Young Water Professionals Conference, Madrid, Spain.*

Under revision:

[8] Pascual-Pañach, Josep, Sànchez-Marrè, Miquel, Cugueró-Escofet, Miquel Àngel, Temporal Case-Based Reasoning for Intelligent Environmental Decision Support Systems. *Submitted to Engineering Applications of Artificial Intelligence in January 2022. In a second revision stage in from October 2022*

In preparation:

[9] Pascual-Pañach, Josep, Sànchez-Marrè, Miquel, Cugueró-Escofet, Miquel Àngel, Case-based reasoning method for online time-series data validation and imputation. *In preparation. To be sent in the forthcoming weeks to the journal IEEE Transactions on Knowledge and Data Engineering (JIF 8.9).*

1.5. Contributions

The main contributions of this thesis are:

- The proposal of a **novel hybrid and general framework for the design of IDSS** for real world processes. This contribution aims to solve a common challenge when designing and implementing decision support tools, which is the lack of a useful and systematic approach to interoperate different models at different stages of the design, as well as to automate the building of the IDSS.
- The proposal of a **temporal CBR approach** to improve the performance of the classic CBR. In dynamic domains, a state at a particular time depends on past states. It means that the identification of situations occurring in a process may not be very accurate when using single cases. The use of episodes instead of cases should improve the identification of similar situations in the retrieval stage.
- The proposal of an **online validation and imputation method based on CBR for time-series**. First, the optimization of a CBR approach is proposed in order to minimize the error between the real value and the estimated one. Then, the performance of the CBR method is compared with other classical techniques, such as Artificial Neural Networks (ANN) and Linear Regression (LR). Finally, different ensemble approaches of all these individual models are proposed to provide a more robust and reliable solution.

2. Background and State of the Art

2.1. Introduction

In this chapter the state of the art and background regarding the main topics and methods addressed in this thesis are presented: Intelligent Decision Support Systems, as the main engineering problem described in Section 2.2, and Case-Based Reasoning (CBR) and Rule-Based Reasoning (RBR) methods, as the tools proposed to tackle that problem, from Section 2.3 to Section 2.5. Finally, in Section 2.6 the domain where the developed methods are deployed and tested is described.

2.2. Intelligent Decision Support Systems

Intelligent Decision Support Systems (IDSS) are Decision Support Systems (DSS) integrating at least one AI technique, and usually combined with other kind of models (e.g. statistical, numerical, mechanistic) to improve the reliability to support the user. Many definitions can be found in the literature. Gottinger and Weimann, 1992 said that "An Intelligent Decision Support System (IDSS) is an interactive tool for decision making for well-structured (or well-structurable) decision and planning situations that uses expert system techniques as well as specific decision models to make it a model-based expert system (integration of information systems and decision models for decision support)". Fox and Das 2000 wrote that "A DSS is a computer system that assists decision makers in choosing between alternative beliefs or actions by applying knowledge about the decision domain to arrive at recommendations for the various options. It incorporates an explicit decision procedure based on a set of theoretical principles that justify the rationality of this procedure". Phillips-Wren et al., 2009 wrote that "IDSS add artificial intelligence (AI) functions to traditional DSS with the aim of guiding users through some of the decision-making phases and tasks or supplying new capabilities". Sànchez-Marrè, 2022 proposes the following definition: "An IDSS is a highly reliable and accurate computer-based system that commonly uses several multidisciplinary methods, either data-driven or model-driven, being at least one of them from Artificial Intelligence field, to support and to improve the decision-making process of a user or users through analytical, synthetic and prognosis tasks in an unstructured complex domain, and being able to learn from past decisions".

When these systems are applied to an environmental domain are called Intelligent Environmental Decision Support Systems (IEDSS), or just Environmental Decision Support Systems (EDSS), according to some authors. Thus, the focus of this research is mainly on this type of applications, although it would be valid for other domains. EDSS are intelligent information systems designed to reduce the time in which decisions are made, whilst improving the consistency and quality of these decisions (Poch et al., 2004). In the last decade, EDSS has emerged as a suitable software tool for decision-making support in order to maximize the performance of the controlled system and to minimize the negative impact of faults on the environment (Cortes et al., 2001).

EDSS field has been trying to use models of the real world to get insight of the behaviour and evolution of the corresponding real systems. Historically, former EDSSs were using only *mechanistic models*. Notwithstanding, usually huge amounts of data gathered from the system being managed were available, hence new empirical models started to be used. *Empirical models* are based on direct observation, measurement and extensive data records. The first empirical models used were mathematical and statistical methods e.g., Multiple Linear Regression (MLR) models. The success of several inductive *machine learning techniques* within the AI area led to their application in EDSSs. Some instances of this usage are e.g., the Association Rules (AR) models, Classification Rules (CR) models, Decision Tree (DT) models or Bayesian Network (BN) models. Since the 80s, both the former mathematical/statistical empirical models and the later machine learning empirical models have been named *data driven methods*, because they result from a mining process using these data. With the use of *data mining models* within the AI framework, the EDSS have evolved to Intelligent Environmental Decision Support Systems (IEDSSs) (Rizzoli and Young, 1997; Sànchez-Marrè et al., 2004; Sànchez-Marrè *et al.*, 2006; Chen et al., 2008; Torregrossa et al., 2017; Nadiri et al., 2018; Han et al., 2020; Godo-Pla et al., 2020; Nawaz et al., 2022). Jakeman et al., 2006 give an insight on the development and evaluation of environmental models.

IEDSSs may be built using a single AI model or integrating several AI models in order to be more powerful, together with Geographical Information Systems (GIS), mathematical or statistical models, environmental/health ontologies or some extra economic information. IEDSSs integrate knowledge/data stored by human experts through years of experience in a certain environmental process operation and management. In addition, knowledge/data can be mined through the intelligent analysis of available large databases coming from historical operation of this environmental process. Thus, *knowledge/data mining model production*, as well as *reasoning* and *interoperation* among the models produced, are key steps to build reliable IEDSSs. The set of AI techniques used ranges from Knowledge-Based Systems (Flanagan, 1980; Berthuex et al., 1987; Maeda, 1989; Gall and Patry, 1989; Tzafestas and Ligeza, 1989; Serra et al., 1994; Ahmed et al., 2002; Aulinas et al., 2011; Castillo et al., 2016; Corominas et al., 2018; Oprea and Dunea, 2010; Oprea 2018), Fuzzy Control Systems (Czoagala & Rawlik, 1989; Wang et al., 1997; Ruano et al., 2010; Santín et al., 2018; Bernardelli et al., 2020), ANN (Capodaglio et al., 1991; Kosko, 1992; Côte et al., 1995; Syu and Chen, 1998; Hamed et al., 2004; Ráduly et al., 2007; Mustafa et al., 2021; Wang et al., 2022), Case-Based Reasoning (Sànchez-Marrè et al., 1997, 2002, 2005; Corchado and Lees, 2001, Fdez-Riverola and Corchado, 2004), Genetic Algorithms (Karr, 1991; Béraud et al., 2007) or Reinforcement Learning (Hernández and Gaudioso, 2011).

Given the plethora of models that might be needed in an EDSS, it is important to reuse existing knowledge by assembling complex models that must interoperate. *Interoperability* is defined as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged” (IEEE, 1990). Additionally, *semantic interoperability* is achieved when the components share a common understanding of the information model behind the data being interchanged (Manguinhas, 2010; Ouksel and Sheth, 1999; Santos et al., 2021). Semantic integration and interoperation have been the focus of some research works in the environmental modelling field, e.g., pioneer work in semantic integration of environmental models for application to global information systems and decision making, specially related to GIS components and models (Mackay, 1999; Wesseling et al., 1996). In addition, some work related with model and data integration and reuse in EDSSs may be found in (Rizzoli et al., 1998), and an overview of model integration is presented in (Argent, 2004). An interesting work in this area is also presented in (Sottara et al., 2012), where the Drools Rule-based integration platform (Salantino, M., et al., 2016) is used as a unified data model and execution environment, and in (Sànchez-Marrè, 2014), where a general framework for the development of interoperable IEDSSs was proposed. In the information systems field several recent works are carried out in semantic integration of business components (Elasri and Sekkaki, 2013; Kzaz et al., 2010). Further works are focused on the semantic interoperability through service-oriented architectures (Vetere and Lenzerini, 2005). Several works have been also developed in the medical domain e.g. (Komatsoulis et al., 2008), where service-oriented architectures were used, or in (Dolin and Alschuler 2011), aiming at data interoperability on Health Level Seven's standard (HL7, http://www.HL7.org). Regarding this issue, one of the most effective ways to interchange information between several software components and share the corresponding information semantics is via XML (eXtensible Markup Language). XML is a meta-language intended to supplement HTML's presentation features with the ability to describe the nature of the information being presented (Erl, 2004). XML adds a layer of intelligence to the information being interchanged, providing meta-information, which is encoded and embedded as self-descriptive labels for each piece of text in the document. XML is implemented as a set of elements, which can be customized to represent data in unique contexts. A set of related XML elements can be classified as a vocabulary. An instance of a vocabulary is an XML document. Vocabularies can be formally defined using a schema definition language like Document Type Definition (DTD) or XML Schema Definition Language (XSD). Furthermore, the Data Mining Group (DMG, 2014) is an independent, vendor led consortium that develops data mining standards, such as the Predictive Model Markup Language (PMML). PMML is a standard for statistical and data mining models, supported by over 20 vendors and organizations. PMML uses XML to represent data mining models. The structure of the models is described by an XML schema. One or more mining models can be contained in a PMML document. A PMML document is an XML document with a root element of type PMML. A PMML document can contain more than one model, and most common data mining models are supported, e.g., AssociationModel, RegressionModel, TreeModel, RuleSetModel, NeuralNetwork, ClusteringModel.

2.3. Case-Based Reasoning

The CBR approach tries to solve new problems in a domain reusing the previous solution given in the past to a similar problem in the same domain (analogical reasoning). Thus, the solved problems constitute the "knowledge" about the domain. As more experienced is the system better performance achieves, because the experiences are stored to be used to solve future problems. This way the system is continuously learning to solve new problems (Riesbeck and Schank, 1989; Kolodner, 1993; Richter and Weber, 2014). It is based on the theory of dynamic memory of Roger Schank (Schank, 1982), which states that the human memory is dynamic and change with its experiences along their lives. Humans learn new things and forget others. The acts of humans are recorded as scripts in their memory. In addition, the processes of learning, understanding, reasoning and explaining are intrinsically bind together in human memory. CBR systems improve their performance becoming more efficient by remembering old solutions given to similar problems and adapting them to fit a new problem rather than having to solve it from scratch. This, in fact, augments the ideas about the components of expertise (Steels, 1990) using the solved cases as an episodic memory: the memorisation of problem solved episodes allows methods to be integrated since they require accessing the past experience to improve the system performance. Also, case-based reasoners become more competent in their evolution over time, so that they can derive better solutions when faced against less experienced situations, preventing them to repeat the same mistakes in the future (learning process).

When working with a CBR system some design decisions have to be considered, such as how to structure cases and its organization in a knowledge base, how to deal with missing information or how to assess the similarity between cases.

As mentioned in Section 2.2, CBR techniques are widely used in the field of decision support on environmental systems, but also in other domains, such as the financial domain (Pérez-Pons et al., 2023), or other applications, such as fault detection (Nasiri and Khosravani, 2019), planning (Jiang et al., 2019) or classification (Gao and Gao, 2021).

2.3.1 Organization and Representation of cases

CBR is based in collecting relevant cases or experiences in a particular domain in order to solve a problem by analogy. Stored cases consist of a description of the experience and the solution provided to that experience. The set of stored cases or experiences can be named the Case Base (CB), or the Case Library or the Case Memory. How sets of cases can be organised, as well as case structures are described below.

The main memory organizations can be divided into two groups: *flat memories* and *hierarchical memories*. In *Flat memories* a set of most similar cases to the input one is retrieved. The main advantage of this approach is that adding new cases to memory is cheap. In counterpart the retrieval time is expensive since every case in the memory is matched against the current case, commonly using a Nearest Neighbour algorithm (Watson, 1996). An example of flat memory is shown in (2-1). Here, one case describes a car using a set of features such as type, brand, number of seats, power, fuel type, colour and price.

Type, brand, seats, power, fuel, colour, price
Car 1: Familiar, BMW, 5, 250, gasoline, black, 45.000
Car 2: Familiar, Renault, 7, 150, diesel, white, 32.000
Car 3: Familiar, Opel, 5, 140, diesel, red, 34.200 (2-1)
Car 4: SUV, Toyota, 5, 150, gasoline, blue, 41.200
Car 5: SUV, Nissan, 5, 190, diesel, grey, 33.500
…

On the other hand, using *hierarchical memories* provides a more efficient retrieval time since only few cases are considered for similarity assessment purposes, after a previous discriminating search in the hierarchical structure. The main disadvantages in this case are the maintenance of the library organization, and the possibility of missing some optimal cases if the search is done in the wrong area. This later problem specially becomes hard in prioritised discrimination networks/trees. Figure 2.1 shows an example of hierarchical memory representation.

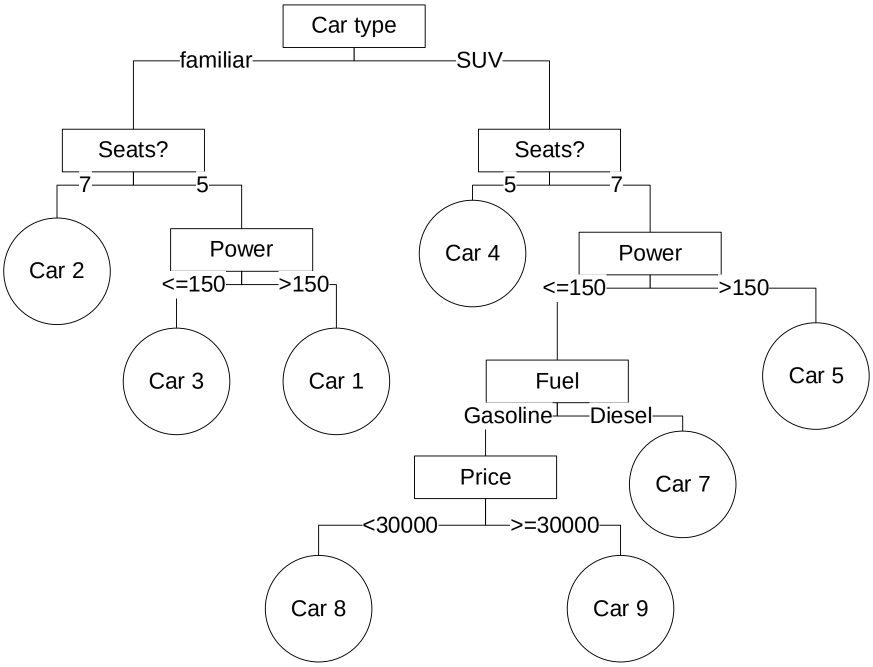


Figure 2.1 Example of hierarchical memory organization

Cases in the CB are learned from real experiences in such a way that they can be reused to solve future problems. Each case can integrate a set of features such as an identifier, a description, a diagnostic, a solution, a utility measure indicating its usage in the resolution of past situations or other relevant information.

2.3.2 Case-Based Reasoning Cycle

The CBR system structure showed in Figure 2.2 considers the four stages of the Case-Based Reasoning method: retrieval, reuse, revision and retain (Aamodt and Plaza, 1994; De Mantaras et al., 2005). The *retrieval phase* is the process by which similar problems (i.e., cases) to the new problem are searched in the CB. Then, in the *reuse phase,* the solution of the retrieved case is adapted and used to solve the new problem. The *revision phase* is to determine whether the solution found in the reuse stage has been successful or not. Finally, the *retain phase* is the stage where useful information from the new problem-solving episode is learnt into the existing CB. The system can learn from both, from successful solutions and from failed ones (repair).

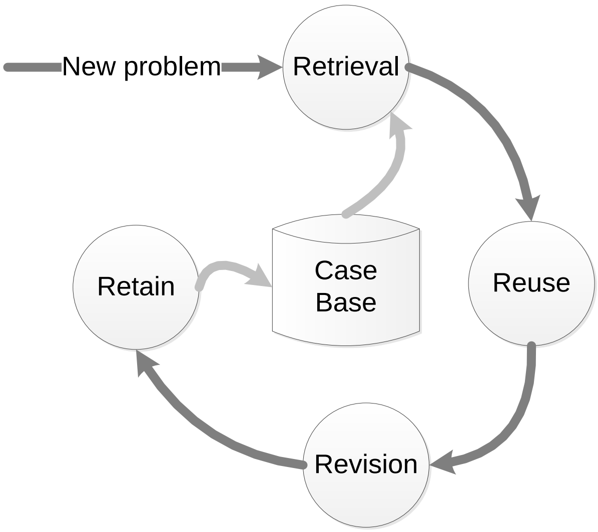


Figure 2.2 Case-Based Reasoning cycle scheme

2.3.3 Case Retrieval and Similarity Assessment

The retrieval task in the CB is slightly more complex than the retrieval in databases. Database retrieval algorithms use exactly matching methods, whereas in a CB, partial-matching methods are used. The aim of the retrieval strategy is to maximise the similarity between the current case and the retrieved one(s). This retrieval process of a case or a set of cases strongly depends on the CB organisation. In flat memories the retrieval time is proportional to the size of the case library, so the performance is penalised by the CB size. Hierarchical memories are more effective in time retrieval because only a subset of cases is considered for similarity assessment after a previous discriminating search in the hierarchical structure. On the other hand, optimal cases are not found because a wrong area is explored.

The retrieval process in hierarchical case libraries usually consists of two main sub steps:

- *Searching* the most similar cases to the new one: the goal of this stage is to find the most promising cases, exploring the hierarchical structure on the basis of some direct or derived features of the new case used as indexes into the CB.
- *Selecting* the best case or cases: the best case(s) among those found in the previous step are selected. Typically, this selection is made by means of a case ranking process using a similarity or distance function. The final retrieved cases are the ones closest (most similar) to the new case.

Selecting the most similar case(s), it is usually performed using some functions to evaluate the distance between cases. The algorithms usually used to compute the distance between cases are nearest neighbour (NN or k-NN) algorithms (Watson, 1996). The partial matching thought each case feature is usually combined in an aggregated partial matching for the whole case. Each feature or attribute is usually assigned an importance value (weight), which is considered in the evaluation function. Depending on the application, feature weights could be static or dynamic.

Most Case-based Reasoning systems such as REMIND (Cognitive, 1992), MEDIATOR (Kolodner and Simpson, 1989), PERSUADER (Sycara, 1987), etc., use a weighted distance function (NN) such as in Equation (2-2):

$$dist(Ci, Cj) = \sum_{1}^{n} w_k \cdot atr_{dist(C_{i,k}, C_{j,k})} \qquad (2\text{-}2)$$

where Ci and Cj are two cases, w_k is the weight for each feature or attribute and the function $atr_{dist(C_{i,k}, C_{j,k})}$computes the distance for feature or attribute k in cases i and j.

Some others measures have been defined in the literature, such as in (Leake et al., 1997; Osborne and Bridge, 1998; Sànchez-Marrè *et al*., 1998; Warren *et al*., 1998).

At this stage, there are two possibilities in a CBR system:

- The situation is *unknown*, i.e., the distance to the most similar case(s) is too big; in other words, there is no memory about this situation.
- There are multiple solutions to solve the same problem, i.e., different solutions can be applied to the current case with the same degree of confidence.

In these situations, CBR systems can be complemented with general expert knowledge coded into the system or can generate an alarm to request the user attention. Other approaches such as in NOOS (Arcos & Plaza, 1995) generate a reflexive task whose goal is to solve that impasse.

2.3.4 Case Adaptation

When the most similar case found in the CB does not match perfectly with the current one, the old solution may be adapted to solve more accurately the new situation. This reusing process can be done during the adaptation process, but also after some feedback from the evaluation step pointing out a problem that should be fixed (repair).

There are a lot of strategies that have been used in the Case-Based Reasoning systems. All these techniques can be grouped (Kolodner, 1993; Riesbeck and Schank, 1989) as *null adaptation*, *structural adaptation* and *derivational adaptation*, although in most Case-based Reasoning systems a mixture of different approaches is implemented.

In *Null adaptation* method, the old solutions are applied directly to the new case. This strategy is often used with simple actions –e.g., accept/reject, a fault diagnosis– such as the first adaptation method used in the PLEXUS system (Alterman, 1988).

In *structural adaptation* methods the adaptation process is directly applied to the solution stored in a case. The structural adaptation methods can be divided in three major techniques: *substitution methods, transformation methods* and *special-purpose adaptation heuristics* or *critic-based adaptation methods*.

- *Substitution methods* provide the solution to the new case by modifying the retrieved solution using the differences between the values of the retrieved case and those ones of the new case in order to guide such modification in the appropriate direction. This approach has been used, for example, in HYPO (Ashley, 1990) and PERSUADER (Sycara, 1987), JUDGE (Bain, 1986). Another kind of methods, such as direct reinstantiation used in CHEF (Hammond, 1989), local search used in JULIANA (Shinn, 1988), PLEXUS and SWALE (Kass and Leake, 1988), query memory used in CYRUS (Kolodner, 1985) and JULIANA, specialised search used in SWALE, etc., can be named as abstraction and respecialization methods. When there is a component as an object or a value, etc, of the retrieved solution, that does not fit in the new problem, these methods look for abstractions of that component of the solution in a certain knowledge structure (concept generalisation tree, etc.) that do not have the same difficulty; the last kind of substitution methods is the *case-based substitution methods*. They use the differences between the new and the retrieved case to search again cases from the case library to eliminate these differences. These techniques have been used, for instance, in systems such as CLAVIER (Hennessy and Hinkle, 1992), JULIA (Hinrichs, 1992), CELIA (Redmond, 1992), etc.
- The *transformation methods* use either some common-sense transformation rules such as deleting a component, adding a component, adjusting values of a component, etc.), as in JULIA system, or some model-guided repair transformation techniques based on a causal knowledge, such as in KRITIK (Goel and Chandrasekaran, 1992) or CASEY (Koton, 1989) systems.
- The *special-purpose adaptation* techniques or *critic-based adaptation methods* are based on some specific rules of repairing, called critics (Sacerdoti, 1977; Sussman, 1975), like those used in PERSUADER. Other systems such as CHEF and JULIA use some domain specific adaptation heuristics and some structure modification heuristics.

Derivational adaptation methods do not focus on the original solution, but on the method that was used to derive that solution. The goal is to use the same method applied to obtain the old solution in order to recalculate the solution for the new case. This methodology was first implemented in ARIES system, and was named as derivational replay (Carbonell, 1986). In such a technique, reinstantiation occurs when replacing a step in the derivation of the new solution, like in systems such as PRODIGY/ANALOGY (Veloso and Carbonell, 1993), JULIA (Hinrichs, 1992) or MEDIATOR (Kolodner and Simpson, 1989).

2.3.5 Case Evaluation

This step provides the Case-Based Reasoning system a way to evaluate its decisions in the real world, allowing it to receive feedback that enables it to learn from success or failure.

Evaluation can be defined as the process of assessing the performance of the proposed solution for the new case adapted from the retrieved one. In non-real time domains, the evaluation process can point out the need for additional adaptation –usually called repair– of the proposed solution. Typically, the evaluation step can be performed either by asking to a human expert whether the solution is a good one or not, by simulating the effects of the proposed solution in the real world or by directly getting feedback on the results of the proposed solution from the real world.

2.3.6 Case Learning

Learning is an essential task of the CBR systems. Mainly, there are two major kinds of learning: *learning by observation* and *learning by own experience*. *Learning by observation* happens when the system is provided with a set of initial cases, given by an expert or by direct observation of real data. Also, an expert can provide a new case in any moment.

When historical data is available, this initial set could be obtaining by applying some classification procedures to historical databases. See for example (Sànchez-Marrè *et al*., 1997). In order to have a complete representation of the particular domain, some objects (cases) belonging to each discovered class are selected to be included in the initial Case Library.

Learning by own experience is the process executed after each reasoning cycle. After the evaluation step, the system can increase its problem-solving capabilities by learning from the last experience. If the proposed solution has been a successful one, the system can learn from it if it is stored in the memory (CB). Then, when new similar case to this one appears, it can be solved in the same way (learning from success). On the other hand, if the solution was wrong, the case can be also learned to prevent future mistakes in the same situation (learning from failure). A CBR system cannot necessarily implement both kinds of learning approaches.

The task of including new cases in the CB should be done taking into account some considerations. First, it should be inserted in the appropriate place in the CB, depending on its organization structure, i.e., flat memory or hierarchical memory, it is important to guarantee that new experiences will be found when necessary. And then, it is also important to avoid the indiscriminate learning. Learning redundant experiences, i.e., too similar ones, can negatively affect the retrieval performance, especially in flat memories organizations.

Regarding the failure possibility, and considering that the evaluation process is correct, there are two reasons that could originate a failure: the best case is found (the most similar one), but it is not similar enough; or although a very similar case exists in the CB, it is not found. In the first case the problem is that there are not enough cases in the CB to cover the whole space of situations. Here the solution is to learn more relevant experiences. In the second case, there is something wrong in the retrieval process, e.g., the distance function is not suitable to capture the difference between features values or the weights are not correctly assigned, or maybe the discrimination attributes used to search through the hierarchical structure are not good; then the algorithm is searching in the wrong place. The solution to this problem relies in re-organising the case library's structure (Veloso & Carbonell, 1993).

To deal with failure situations there are several possibilities. The first one is storing it in the CB in order to avoid future errors. Some case-based systems maintain a separate case library of failed cases (Hammond, 1989), and others maintain only one case library structure. In the first type of system, and before retrieving any successful case, a previous phase to find any similar failed case is executed. Depending on CB structure, these possible failed cases are retrieved from the failed cases CB or applying a filter to search in the unique CB. In any case, failed solutions can be avoided. A second interesting possibility is to incorporate a human expert to the reasoning cycle, who can introduce the right solution to the new situation. So, in future similar experiences, the case can be used properly. Finally, other actions that can be performed include updating feature weights to modify the distance function (Bareis, 1989; Koton, 1989), or in the case of hierarchical structures, modifying features order in the discriminating list. Another feature is to update the utility measure of the retrieved cases that could derive the new case.

2.4. Temporal Case-Based Reasoning

In the previous Section 2.3 the CBR technique is presented. In classical CBR approaches, knowledge is codified using static cases describing a situation at a particular time instant. But temporal or dynamic domains suggest that not only the description at a particular time instant is important, but also how that situation has been reached. Thus, considering relations between consecutive states may be interesting for increased accuracy in the identification of similar situation, i.e., retrieval accuracy. Assuming this idea, there are some works that proposes methods in order to consider the temporal component in CBR-based systems. Since the topics and applications are very diverse, they are grouped according the topic or target problem to solve, such as planning, prediction and control/supervision or decision support and monitoring. Regarding planning topic, Ram and Santamaria, 1997 proposes a new method for continuous CBR and discusses its application to the dynamic selection, modification and acquisition of robot behaviours in an autonomous navigation system. In Navarro et al., 2012, the use of a temporal bounded case-based planning approach is proposed for industrial processes, planning to solve problems with temporal constraints. This approach is similar to the one presented in Montani and Portinale, 2005; the authors propose a framework to represent temporal information in two levels, the case level and the history level. It is applied to RHENE, a system for managing patients in haemodialysis regimen. Ma and Knight, 2003 proposes a framework for Historical Case-Based Reasoning (HCBR), allowing the representation of temporal knowledge. The need of dealing with the history of cases is illustrated using a planning problem. Concerning the prediction topic, Branting and Hastings, 1994 describes CARMA, a system for rangeland pest management advising. This system integrates model-based reasoning and case-based reasoning for prediction in rangeland ecosystems, using a method called temporal projection. In Jaere et al., 2002, a methodology for representing temporal knowledge in a qualitative and interval-based framework inspired on Allen's interval logic (Allen 2013), is presented. The latter approach is applied to the prediction of unwanted events in oil well drilling. In Schmidt and Gierl, 2002, the combination of temporal abstraction and case-based reasoning is used for prognosis in medicine. The authors point out the importance of reasoning about time in medicine –e.g., to describe the tendency about the status of a patient or in diseases fore-casting. Some works address the control/supervision or decision support issue. The proposal in Jaczynski, 1997 is based on a method called time-extended situations. The method proposes to use not only the current state of the observed process, but also its past history. It is illustrated on a plant nutrition control system. Sànchez-Marrè et al., 2005 suggests that cases can be grouped in sequences of cases making up episodes. Here, a framework for an episode-based reasoning approach is presented. Brown et al., 2018 proposes the use of temporal sequences to consider preceding events when recommending solutions, rather than looking at events in isolation. The method is applied to a Type I Diabetes Mellitus Bolus Insulin decision support system. In Meléndez et al., 2000 a similar approach to the one in Martí Navarro et al., 2012 is applied to the control of sets of recipes – forming a temporal problem– for making products like plastic or rubber pieces. Other works deal with monitoring tasks. Martín and Plaza, 2004 proposes Ceaseless Case-Based Reasoning (CCBR), a new CBR model that aims solving problems which involve temporally sequences of observational data. This paradigm could be applied e.g., in intrusion detection systems (IDS). In Lupiani et al., 2017, the authors point out that the inclusion of temporal features within the problems descriptions has gained importance in the last years, and they to apply temporal CBR to smart homes for elderly people monitoring, with cases defined as a set of events representing the activity of the monitored person.

2.5. Rule-Based Reasoning

Experts have gained knowledge from experience about how to solve problems in a specific domain. This expert knowledge can be coded using different knowledge representation formalisms, such as inference rules. As the knowledge is expressed with rules, the reasoning mechanism is named as Rule-Based Reasoning (RBR) or Knowledge-Based Reasoning (KBR) (Jackson, 1999; Buchanan *et al*., 1983). In early RBR systems, also named as Expert Systems, this kind of systems was also named as Production Systems, because the inference rules produce new knowledge.

The architecture of an RBR system is represented in Figure 2.3. The main components of an RBR system are: the fact base, the knowledge base, the inference engines (reasoning component), the meta-reasoning component, the user interface, the knowledge acquisition module and the explanation module, and the knowledge engineer interface.

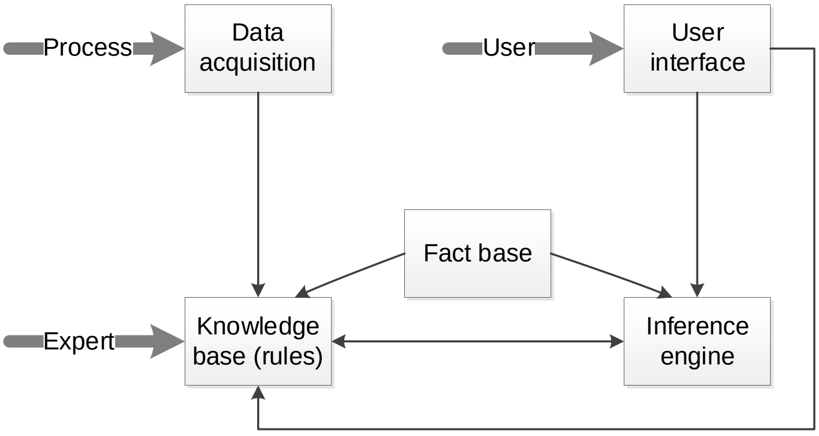


Figure 2.3 Architecture of a Rule-Based Reasoning System including main components

These components will be described an analysed next.

2.5.1 Fact Base

Facts characterize different and relevant situations of a particular domain. Each fact has some static properties and some dynamic properties. Dynamic properties are the ones describing the current state of the problem to be solved.

Usual static properties are:

- An identifier or name of the fact,
- Its type, describing the values that can manage, e.g., boolean, numerical, string.
- How the value of the fact could be obtained, e.g., asking to the user or not, as well as the question to be asked if yes.

The most common dynamic properties are:

- The value of the fact at a given moment of the reasoning process.
- The certainty measure of this value, expressing the degree of certainty of the value at a given moment of the reasoning process. The certainty measure can be expressed using different models, e.g., without certainty, i.e., all values are sure, using the possibility theory (values in the interval [0, 1]) or a qualitative probabilistic label, such as {completely improvable, very improbable, improbable, unknown, probable, very probable, completely probable}.

These dynamic properties have to be consulted often, so facts need to be organised in fast-accessing data structures, such as hash tables. The example below (2-3), from the waste water treatment domain, represents a fact with all its properties:

$$\begin{array}{l} Fact\ id: Ammonium\ concentration \\ Fact\ type: Boolean \\ How\ the\ fact\ is\ obtained: online\ measurement - NH_4 > NH_4^{max} \\ Fact\ current\ value: False \\ Fact\ certainty: Completely\ probable\ (sensor\ status\ ok) \end{array} \tag{2-3}$$

2.5.2 Knowledge Base

The Knowledge Base (KB) is the core of an RBR system. It is the component where the knowledge obtained from experts is stored. As mentioned before, the knowledge is represented using rules.

The rules have the form in Expression (2-4):

IF <conditions> **THEN** <actions/conclusions> (2-4)

Each rule can be composed by several elements:

- A rule identifier assigned to each rule.
- A set of conditions (antecedent) and actions or conclusions (consequent). When conditions are satisfied, the rule is fired and actions can be executed or conclusions deducted. Several conditions can be combined using different logical operators, such as negation (NOT, Table 2.1), conjunction (AND, Table 2.2) or disjunction (OR, Table 2.2).

Table 2.1 NOT operator

<condition1>	NOT condition1
false	true
true	false

Table 2.2 AND / OR operators

condition1	condition2	condition1 AND condition2	condition1 OR condition2
false	false	false	false
false	true	false	true
true	false	false	true
true	true	true	true

Conditions can be expressed using propositional logic or first-order logic. Propositions are statements which can be true or false, e.g., it rains, whereas first-order logic statements contain variables so that the same statement can be expressed with multiple instantiations, e.g., *x* is a fruit –$fruit(x)$. Depending on the value of the subject *x*, the statement could be true or false. Some examples of actions can be sending messages or run some calculations, whereas conclusions will be new deductions about the truth or not of some facts in the Fact Base.

- A rule certainty measure of the co-occurrence of the conditions and conclusions of the rule. The certainty values can be expressed with the same uncertainty management models detailed above for the facts' values.

The example below (2-6), from the waste water treatment domain, represents a rule with all its properties:

$$\begin{array}{l} Rule\ id: Nitrification\ condition \\ Rule\ antecedent: (NH_4 > NH_4^{max} \wedge NO_3 < NO_3^{max}) \vee (NH_{4_{24h}} > NH_{4_{24}}^{max}) \\ Rule\ consequent: Activate\ Blower \\ Rule\ certainty: Completely\ probable\ (sensors\ status\ ok) \end{array} \tag{2-5}$$

2.5.3 The Inference Engine

The *reasoning module* is composed by the *inference engine*. Its main goal is to deduce/demonstrate new facts and to execute actions to solve the current problem, considering a set of facts, a given knowledge base and the user interaction.

The reasoning strategy of the inference engine can be divided in the following steps:

- *Detection*: in this step the applicable rules are detected. Here some conflicts may arise, because usually more than one rule has to be used.
- *Selection*: this step selects to rule to be applied or fired, thus the conflict in the previous stage is solved.
- *Application*: finally, the selected rule is applied and the corresponding actions executed / conclusions reached. This last step is commonly known as the Inference step, because at this step is where new facts are produced to be validated and stored in the Fact Base.

Regarding the Inference engines, there are two main inference strategies:

- *Forward reasoning* or deductive engine: this is a *data-driven* approach. It starts with the data or evidences available in the Fact Base, and tries to apply rules to produce some new conclusions. When the set of antecedents of a rule are found in the Fact Base to be true, or after asking the user, then the rule is fired, which means that the new conclusion obtained can be added to the Fact Base. Therefore, facts are guiding the process, because rules are applied from left to right. An example is introduced in (2-6):

$$A \wedge B \wedge C \rightarrow H \qquad (2\text{-}6)$$

If the conditions of the rule in (2-6) are true, i.e., A, B and C are true, then action / conclusion H can be executed / becomes true, due to the application of logical modus ponens mechanism (A implies H, A is true, then H must be true).

- *Backward reasoning*: this is a *goal-driven* approach. The process tries to validate a conclusion using the rules in the Fact Base that can conclude that hypothesis. This means that conditions appearing in the antecedent of the rules should be also validated. Therefore, the conclusions are guiding the reasoning process, because the rules are applied in from right to left (2-7):

$$H\ IF\ A \wedge B \wedge C \qquad (2\text{-}7)$$

To validate the consequent H, A, B and C are new sub-consequents that should be validated. Thus, it is necessary to find the facts A, B and C in the Fact Base or as consequents in the available rules in the Knowledge Base. In this later case, the reasoning process becomes a recursive reasoning process, while consists in the exploration of an AND/OR graph.

2.5.4 Meta-Reasoning Component

The Meta-reasoning component is in charge of controlling how the reasoning process is done, i.e., when and how to apply the reasoning processes, giving a solution to a conflict situation. A conflict situation occurs when more than one rule can be fired. The meta-knowledge can be implicit or explicit:

- *Implicit Meta-knowledge*: one way of controlling when to use a rule is to impose a particular ordering of the rules' execution by adding some artificial premises that ensures the desired order. One of the major problems with this mechanism is the loss of meaning of the knowledge coded in the rules. As the number of rules increases, its complexity is increased.
- *Explicit Meta-knowledge*: The problem or implicit meta-knowledge is solved by the use of meta-rules (Davis, 1980). A *meta-rule* is a rule acting over rule/s. This mechanism provided a clear separation between *control* and *knowledge*. Rules express the knowledge about some domain, and the meta-rules express the control strategy over the knowledge. The use of meta-rules is fully linked to the use of a modular structure for the Knowledge Base. In addition, it is important to highlight first, that it provides a unified reasoning mechanism, i.e., rule and meta-rules have the same structure; and second, it also provides a kind of strategy for the problem-solving process.

From the semantic point of view, different types of meta-rules can be defined:

- Meta-rules over rules: this kind of rules controls the activation or not of some rules, depending on if they should be considered or not in the problem-solving under particular conditions.
- Meta-rules over modules: These meta-rules have the purpose of addressing the knowledge in different ways: forward or backward reasoning strategy, setting the minimum certainty level of the rules to be applied or rules subsumption.
- Meta-rules over strategies and actuation plans. These meta-rules allow to stablish, for example, which modules and in what order should explored (strategy), which strategy should be applied first in the case of that more than one is possible (action plan) or detecting exceptions in the problem-solving process.

As mentioned before, from the syntactical point of view meta-rules have the same structure than rules, but with some peculiarities. While the antecedent will be a combination of conditions linked with logical operators, the consequent will depend on the type of meta-rules. For example, a list of identifiers of rules to be activated / deactivated, the specification of a reasoning mechanism: forward or backward, in this last case, with the conclusion (fact) to be validated, a list of modules to be explored or the stop of the reasoning process due to an exception. A simple example of a meta-rule, that could be found in the waste water treatment domain, could be the one in (2-8).

IF *it's raining* **THEN** *activate bypass rules* (2-8)

2.6. Application domain: waste water treatment

This section describes a general overview of the domain of application where the general framework and the methods developed in this thesis have been applied.

2.6.1 Integral Water Cycle

The integral water cycle (Figure 2.4) is defined as the trajectory that the water takes from when it is collected in its raw state of nature until its supply to homes, and closing this trajectory with an inverse journey once again back to the environment.

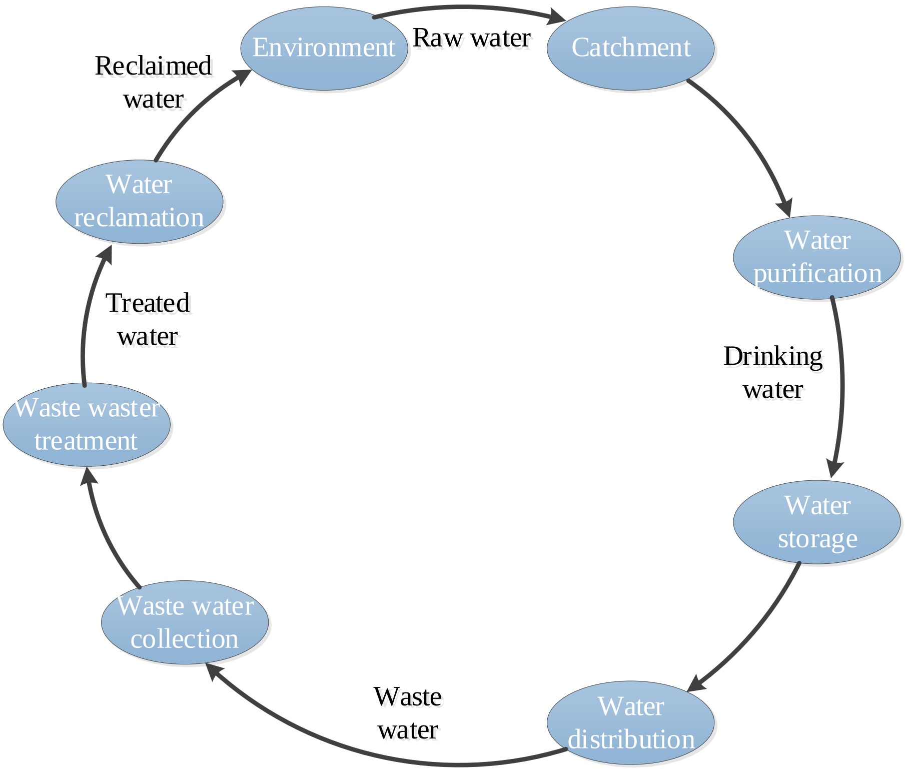


Figure 2.4 Integral urban water cycle

This cycle is composed of six stages:

1. Catchment: The first step is to obtain the required quantity of water from nature (raw water) that is needed not only for human consumption but also other anthropogenic activities (e.g. industrial, commercial). This catchment can be done from surface sources (e.g. reservoirs, the sea) or underground sources (i.e. aquifers).

   In recent years, due to the increased water scarcity, active research is being carried out to achieve the reuse of water for purposes other than human consumption, such as watering green areas or cleaning streets. Thus, water catchment can be done not also from environment but from the result of some reuse processes obtaining reclaimed water.

2. Water purification: Raw water must be treated to be used for human consumption (drinking water). This process is carried out in Drinking Water Treatment Plants (DWTP) using different combinations of technologies, e.g. coagulation, sedimentation, filtration, chlorination.
3. Water storage: Drinkable water obtained from the DWTP is transported to tanks connected to the supply network. Its objective is to ensure a continuous supply of drinking water under some controlled parameters and to secure availability in the future taking advantage of those moments where the produced water exceeds the current needs.
4. Distribution: Water distribution can be divided into two scales: high and low networks. The high network is responsible of taking the water from DWTP to the cities through a network of large and medium diameter pipes. Then, the low network distributes water to end users (e.g. homes, industries) through smaller diameter pipes.
5. Collection: Once distributed water is used, then it is necessary to collect the water that has been discarded (waste water). The sewage network consists of pipes and conduits in charge of collecting and transporting waste water from end user sites to treatment plants. The design of these networks should consider different variables such as the population increase or the terrain orography, in order to estimate the normal speed in each section or the behaviour of the network in the heavy rain events (waste water and rain water share the same network). As in drinking water distribution the network can be divided into the high network (large diameter pipes and conduits from the low network to WWTPs. and the low network (from end users to the high network, using smaller diameter pipes).
6. Treatment: Waste water collected by the sewage network reaches WWTPs (Figure 2.5). The aim of waste water treatment is to obtain water that can be returned to the environment. Waste water goes to a complex process divided into different stages and based on physical, chemical and biological techniques in order to remove and/or reduce contaminants that are potentially dangerous for the environment and human health.

Figure 2.5 Aerial view of a Waste Water Treatment Plant

As mentioned in catchment stage, the advances in waste water treatment technologies provide enough quality water not only to be discharged to the environment but also to be reused for different non-human consumption purposes (reclaimed water).

2.6.2 Industry 4.0 in the water sector

Industry 4.0 is disrupting the way companies manufacture, improve and distribute their products. This revolution is driven by the integration of different new technologies into their production facilities and throughout their operations, such as Internet of Things (IoT), cloud computing and analytics, Artificial Intelligence (AI) and machine learning techniques, cybersecurity or digital twins (Figure 2.6).

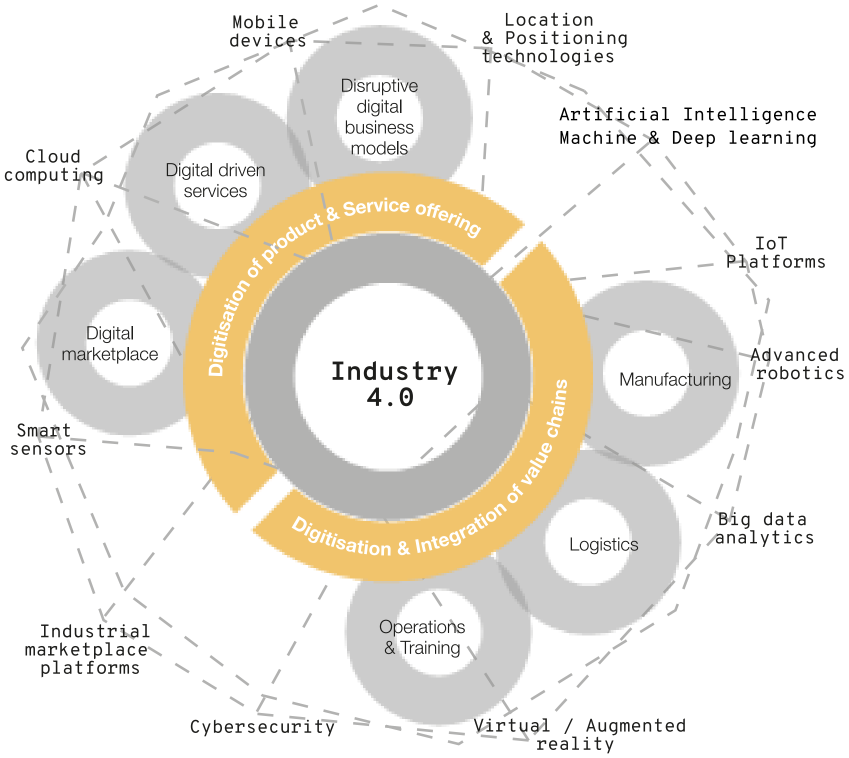

Industry 4.0
Digitisation of product & Service offering
Digitisation & Integration of value chains
Disruptive digital business models
Digital driven services
Digital marketplace
Manufacturing
Logistics
Operations & Training
Mobile devices
Location & Positioning technologies
Artificial Intelligence Machine & Deep learning
IoT Platforms
Advanced robotics
Big data analytics
Virtual / Augmented reality
Cybersecurity
Industrial marketplace platforms
Smart sensors
Cloud computing

Figure 2.6 Industry 4.0 diagram. Source: i2CAT (https://i2cat.net)

Internet of Things (IoT) is one of the key components of this smart factories. The use of sensors with many connectivity capabilities makes it possible the collection of large amounts of data. Cloud computing enables the connectivity and integration across different business units, as well as providing a more efficiently and cost-effectively framework for data storage and analysis. AI and machine learning techniques allow taking advantage of the high volume of information generated. It is precisely this connectivity capabilities what exposes new entry paths for malicious attacks and malware. Thus, cybersecurity is gaining importance within companies' interests. Also, the connectivity and sensorisation of processes allow companies to create digital twins, that are virtual representations of real-world entities and processes, synchronized at a specified frequency and fidelity (Digital Twin Consortium, 2023). These copies use real-time and historical data to represent the past and present and simulate predicted futures.

In the water sector (Cugueró-Escofet and Puig, 2023), the smart water concept is incipiently emerging and strongly depends on properly address several key challenges, e.g., the formulation of an integrated water information system with standardised ontologies in order to achieve real interoperability (Gourbesville, 2016), as suggested e.g., in the Smart Water Management Initiative introduced in (Choi et al. 2016). But although the conceptual solution for an urban information system is today commonly presented as a consensus solution, several technical challenges are still to be tackled in order to achieve real integration and functional interoperability (Gourbesville, 2016), particularly, in the definition of standards for managing workflows among various applications and models to produce useful real time information for decision makers. In (Sànchez-Marrè, 2014) the interoperation and integration of several AI and mathematical models within the same system is presented as one of the main challenges in the research in IDSS. The lack of interoperability standards in Information and Communication Technology (ICT) systems for water management is also pointed out in (Laxmi and Laxmi-Deepthi, 2017; Robles et al., 2014), jeopardizing proper monitoring, control and overall efficiency of water management and preventing their evolution and improvements e.g., the adoption of Internet of Things (IoT) paradigm. The need for standards in the management of water infrastructures is also pointed out in (Di Biccari and Heigener, 2018) as an essential step for a fully integrated management and for reaching efficient levels of interoperability and communication. The special issue Smart Urban Water Networks in Water journal (Di Nardo et al., 2021) collects some papers about crucial topics on Smart Urban Water Networks, such as reliability, resilience, and performance of water networks, innovative demand management, and the novel challenge of real time control and operation, along with their implications for cyber-security. The special issue Advanced Hydroinformatic Techniques for the Simulation and Analysis of Water Supply and Distribution System by (Herrera et al., 2018) some of the problems of water systems are addressed, such as the design of water systems, networks optimization and performance assessment, modelling and forecasting of water demands or optimal management of water quality. In addition to the inherent complexity of water systems, water companies should face the challenge of handling the huge amounts of data generated or the need for online actions to accomplish real time-decision making.

(Poch et al., 2017) points out the gap between the research in this field and the water market, and suggest that the success in the implementation of EDSS lies on putting the focus on the transfer to the market. In (Mannina et al., 2019), a review of the state of the art in DSS for WWTPs is presented. Here, the development of user-friendly applications and the challenge to reach the water market are also emphasized. (McIntosh et al., 2011) presents an overview or the challenges and best practices for the successful development of EDSS. (Elsawah et al., 2020) identifies the eight grand challenges in socio-environmental systems (SES) modelling: bridging epistemologies across disciplines; multi-dimensional uncertainty assessment and management; scales and scaling issues; combining qualitative and quantitative methods and data; furthering the adoption and impacts of SES modelling on policy; capturing structural changes; representing human dimensions in SES; and leveraging new data types and sources. Although here these challenges are presented from the point of view of modelling social and environmental systems, some of them are also relevant for the design of IDSS, such as the combination of different types of methods, or the availability or quality of data.

In the CBT framework and as a result of research in the Decision Support field research, a commercial IDSS —which was used in more than 100 WWTPs around the world— was built in a former stage of this work. This IDSS was initially based on the methodology described in (Poch et al., 2004). They proposed an EDSS architecture based on five levels: data gathering, diagnosis, decision support, suggested plans and actions, as well as the integration of different AI technologies is proposed, including RBR, CBR and ANN with other classical methods (classical control systems or models). Then, as described in (Poch et al., 2017), the IDSS was drastically simplified for its commercial implementation, using only a rule-based component, and did not aim at scalability, dynamic learning and gradual competence increase, interoperation of methods and usability issues.

2.6.3 Waste Water Treatment Plants

The current study, as a particular instantiation of an environmental system where the proposed IDS methodology can be applied, is developed in the framework of real WWTPs managed by CBT. Plants capacity ranges from 1000 m^3/day to 40000 m^3/day, including water and sludge lines, and in some cases, a biogas line. Despite the similar layout among CBT WWTPs, there are some particularities that imply a custom-made control system, e.g., number and type of actuators and sensors or influent characteristics.

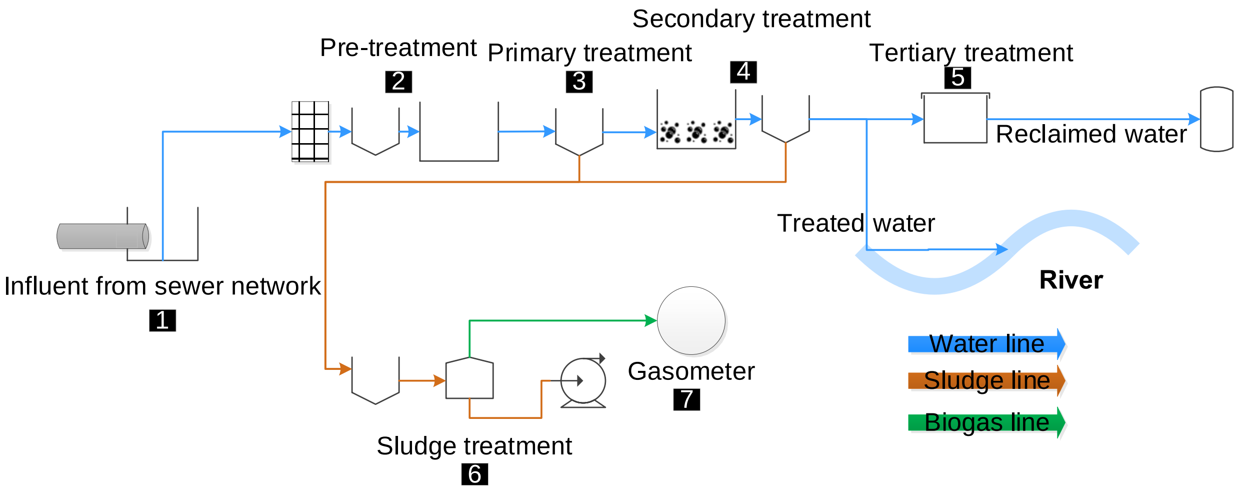


Figure 2.7 WWTP generic scheme

Figure 2.7 shows a generic scheme of a WWTP including the most common treatment operations:

1) Waste water collection through the sewer network.
2) Pre-treatment, with the objective of removing coarse solids and other large materials, as well as sand and oil or grease found in waste water. This stage is important to enhance the operation and maintenance of subsequent treatment units.

3) Primary treatment: removal of settleable organic and inorganic solids by sedimentation, and the removal of floating materials by skimming.
4) In the secondary treatment microorganisms are used to biologically remove organic matter and nutrients from waste water. CBT WWTPs are based on the activated sludge process. There is a large variety of configurations, but all of them consist of three main components: a bioreactor, which can be divided in aerobic and anoxic zones, a secondary clarifier and a sludge recirculation from the clarifier to the bioreactor. The output of the secondary treatment is the treated water, which is discharged to the environment. The quality of the WWTP effluent is regulated by the directive 91/271/EEC. The aeration of the biological reactor in the activated sludge treatment is the most resource consuming process, accounting for about the 50 % of the overall treatment process energy use of the WWTP (Feng et al., 2012; Oulebsir et al., 2020). Thus, it one of the most critical processes to be correctly supervised and controlled (Olsson et al., 2015)
5) The tertiary treatment is used to remove compounds that are not removed in the secondary treatment. For uses other than direct discharge to the river, i.e., recreational uses, irrigation, street cleaning, etc., water requires an upgrade of the water quality acceptable for reuse, obtaining reclaimed water. In Spain, the RD 1620/2007 stablish the legal framework for the reuse of treated water, defining the possible uses, quality and analysis requirements. The access to adequate water supplies is essential for human development. It is expected that water scarcity problems will be aggravated by climate change. Thus, recycling water for non-drinking uses is being an important adaptation measure to save resources.
6) Sludge is a by-product of waste water treatment. It contains organic and inorganic matter, nutrients, chemicals and other pathogens, so it is extremely important to properly treat it to minimise its environmental impact. The sludge treatment usually includes three steps: thickening, anaerobic digestion and dewatering. The sludge thickening is a mechanic process to reduce the sludge volume removing the excess of water that it contains. The thickening can be done by gravity (sludge is decanted) or flotation (sludge floats). In the anaerobic digestion process microorganisms break down the organic matter contained in the sludge and convert it in biogas, composed mainly of methane and carbon dioxide carbon, which can be used, for example, to produce electricity or heat. Finally, the remaining sludge is dewatered before its final disposal. Depending on its sanitization level it can be used as a fertilizer or discarded to landfill.
7) Biogas produced in the anaerobic digestion process is stored in a gasometer. Currently it is used to produce electricity and fed it into the grid. Other possibilities include the biogas upgrade to be injected in the natural gas network.

With the circular economy model being one of the main blocks of the European Green Deal, it becomes increasingly important to produce added-value products from the waste water treatment, such as reclaimed water, nutrient recovery or biogas production (Bakan et al., 2022)

2.7. Conclusions

In **Chapter 2** a review of the state or the art regarding the main topics and methods involved in the development of the work presented in this thesis are discussed. Tools and methods used are described in Sections 2.3 (CBR) and 2.5 (RBR). Section 2.4 presents some works about how to consider the temporal component in CBR when it is applied to dynamic domains. In view of the referenced works, it can be highlighted the importance of the time component in many different domains, as well as the fact that there is no single solution to integrate the time component in the classical CBR method. Regarding the application domain, wastewater, or the water sector in general, it is confirmed that the definition of standards and the interoperability and integration of different AI methods and applications is one of the current challenges to be faced. Finally, it is also emphasized the gap between research and the water market, which is also one of the challenges addressed in this work.

3. An IDSS Methodology for Intelligent Process Control Systems

3.1. Problem description

The first step before going into detail on the methodology is to define the problem to be solved. The supervision and control solution for a particular process, depends on the characteristics of this process. Classical supervision and control tools are ad-hoc solutions for a particular system, or at least need to be explicitly configured and parameterised. This fact involves a lot of time and resources in the design stage but also in start-up stage and maintenance tasks. In the same way, required modifications due to changes in the process —e.g., new sensors or actuators— or user demands may also entail time-consuming changes in the code of the application. This thesis proposes an IDSS framework (Pascual-Pañach et al., 2018a; 2018b; 2019b; 2021) for deploying Intelligent Process Control Systems and a general efficient methodology based both on data and models, to allow scalability to further types of systems. Furthermore, the IDSS performance is also considered. The use of explainable approaches such as CBR and RBR provides to the user a reliable solution to supervise and control the process, based on reproducing human decisions. Thus, it is expected to obtain., at least, a similar performance than the one obtained before the IDSS implementation, guaranteeing the required quality for treated water by reusing solutions used in the past.

3.2. System architecture

Figure 3.1 shows the integration of the IDSS (coloured in orange) proposed here in the general architecture used for the control and supervision of many processes.

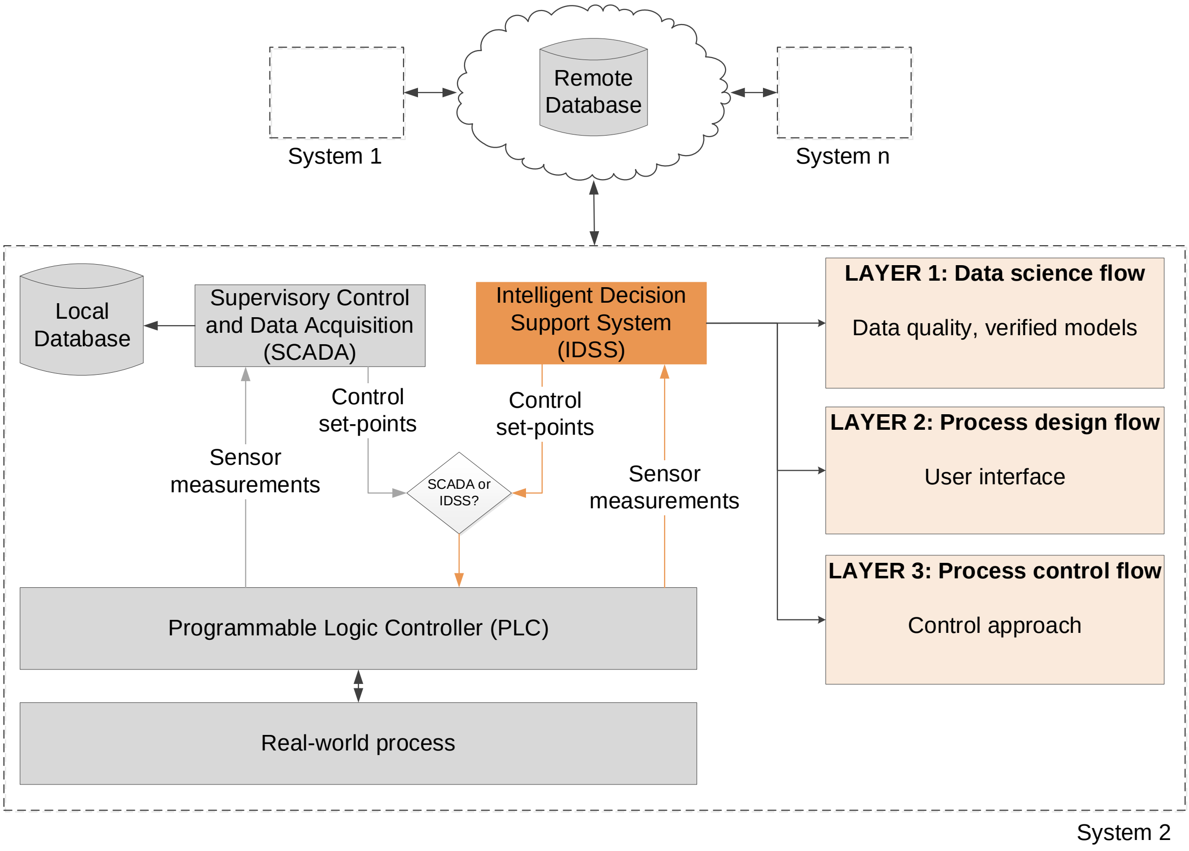

Remote Database
System 1
System n
Local Database
Supervisory Control and Data Acquisition (SCADA)
Intelligent Decision Support System (IDSS)
LAYER 1: Data science flow
Data quality, verified models
LAYER 2: Process design flow
User interface
LAYER 3: Process control flow
Control approach
Control set-points
Control set-points
Sensor measurements
Sensor measurements
SCADA or IDSS?
Programmable Logic Controller (PLC)
Real-world process
System 2

Figure 3.1 Architecture of the proposed Intelligent Decision Support Methodology

The IDSS reads online sensor measurements from the process and generates, through its IPCS component, the set-points for a lower control level device, usually a PLC. A standard control system consists of the combination of a SCADA system and a PLC, such as the particular solution described in Section 2.6. The SCADA is a software system used to control, monitor and acquire data from a process, while the PLC is a modular industrial computer that provides multiple inputs and outputs and contains the control loops programming. Traditionally, SCADA/PLC systems integrate classical control approaches, e.g., PID controllers, although nowadays more complex approaches can be integrated in these systems. The IDSS is based on AI techniques, with the aim of providing a scalable solution to different installations. In the scheme represented in Figure 3.1 both systems –i.e., SCADA/PLC and designed IDSS– can work together in parallel, providing redundancy to the system and including some set-points selection criteria.

The IDSS is based on a three-layer architecture (Figure 3.2).

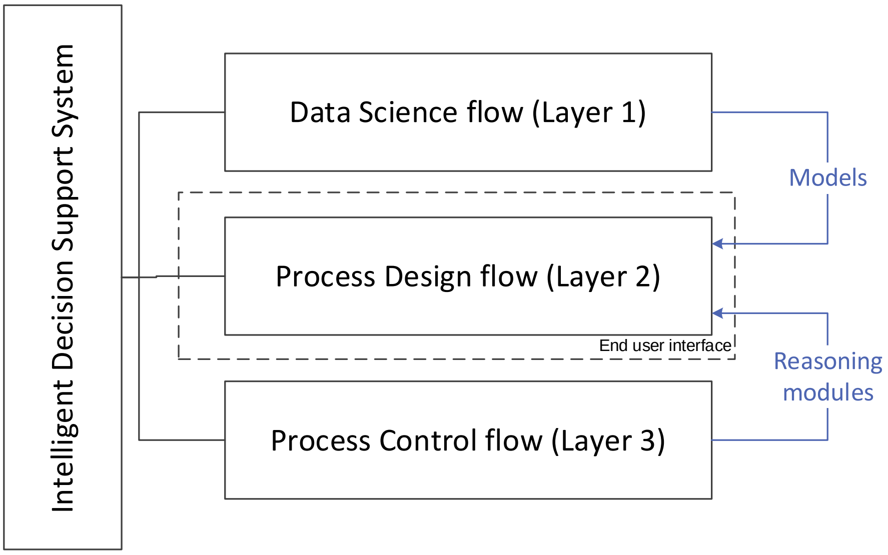

Intelligent Decision Support System
Data Science flow (Layer 1)
Models
Process Design flow (Layer 2)
End user interface
Reasoning modules
Process Control flow (Layer 3)

Figure 3.2 Intelligent Process Control structure

The Data Science flow layer (i.e., Layer 1) is used to generate models obtained from process data or expert knowledge. It is an offline procedure that takes historical available data from each system with the aim of generating valid data-driven models to supervise and control the process. The input of Layer 1 is a standardized and properly formatted raw database containing all available data for each system, namely: sensor measurements, equipment states and alarms, process set-points and further data derived from them. First, different data validation and reconstruction methods, such as the ones in (Cugueró-Escofet et al., 2016; Gibert et al., 2010, 2018a; De Mulder et al.,2018), can be applied to obtain a new filtered and valid database. Then, data mining methods can be applied to discover valid models to be used in Layer 3. These models can be e.g., rule models induced from decision trees or case bases. These models can interoperate in Layer 3 to supervise the system by diagnosing the process status, e.g., normal operation or different abnormal situations, and also to control the process by generating actuator set-points based on knowledge obtained from data. Rule models can also include human expert knowledge of the system (model-driven method), so a user-friendly interface to integrate such human-based knowledge is considered for Layer 1. In the Process Design flow layer (i.e., Layer 2), the layout of the plant is designed, including all processes to be supervised and controlled, and the corresponding signals.

3.2.1 Data Science flow

Data-driven approaches depends on the analysis and interpretation of data to make a decision. Thus, main problems for data-driven methods design are data availability, organization and quality. The process data is stored in the local database of the SCADA system of each process. The first step is to retrieve historical data from the database. At this point is important to consider all possible process behaviours due to, for example, seasonality, i.e., the process behaviour depends on the temperature, as well as recent data representative of the process. Otherwise, obtained models would not be reliable. Once the data is retrieved from the database two main steps are considered to obtain valid data-driven models: data pre-processing and data mining stages.

The first problem to deal with in the first step is the potential huge amount of available data, which could be possibly reduced e.g., by simple resampling according to the process dynamics. Second, as commented previously, the quality of these data is often low due to different causes, namely missing values, inconsistency between values with the same time stamp, out of range values or abnormal behaviours caused by faulty sensors or maintenance operations. Generally, data validation methods are not applied before data storage, or in the best-case scenario are straightforward —e.g., sensor full scale limits. These behaviours suggest the use of a data validation and reconciliation stage during the pre-processing step in order to provide reliable and complete datasets. After detecting problems in the data two options can usually be considered here to process them: 1) to remove these invalid low-quality samples from the data base; 2) to replace invalid values with possible valid reconstructed data using e.g., available temporal/spatial redundancy provided by analytical relations between different variables, sometimes measured at different times (temporal redundancy) and/or locations (spatial redundancy) (Cugueró-Escofet et.al. 2014; 2016). After the data pre-processing process, the production of a valid set of data is assumed.

The second step is the data mining stage. Here, using a valid set of data different kinds of data-based models can be obtained. This thesis is focused on CBR systems complemented with an RBR module based on expert knowledge, as it can be seen in Figure 3.3.

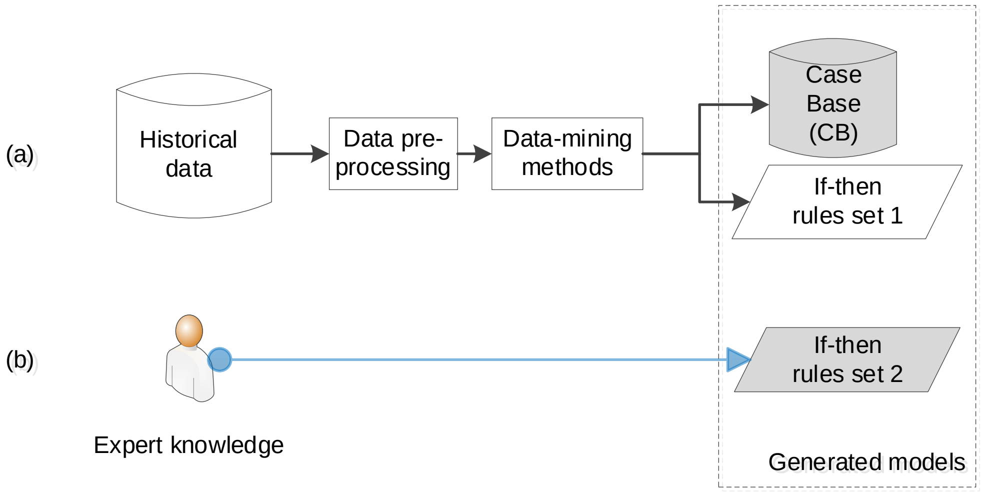


Figure 3.3 Data science/Model layer: a) data-driven, b) model-driven

3.2.2 Process design flow

The Process Design flow layer (Figure 3.4) is the one used as a Graphical User Interface (GUI) for the application. It has two main purposes. First, the configuration of the layout with the process units, including all the measured values and control set-points. This configuration is done by connecting the models generated in Layer 1 and the methods used in Layer 3 with the real elements in the controlled and supervised process, i.e., allows the user to configure the IDSS. For example, a set of rules defined in Layer 1, describing how to manage the setpoints for a particular process, and the RBR module in Layer 3 used to evaluate those rules, are connected in Layer 2 with the corresponding process element. A process element would include different features, such as involved data and a set associated Key Performance Indicators (KPIs). Second, to show to the user useful information about the process, including KPIs values, to analyse the performance of the tool and decision support.

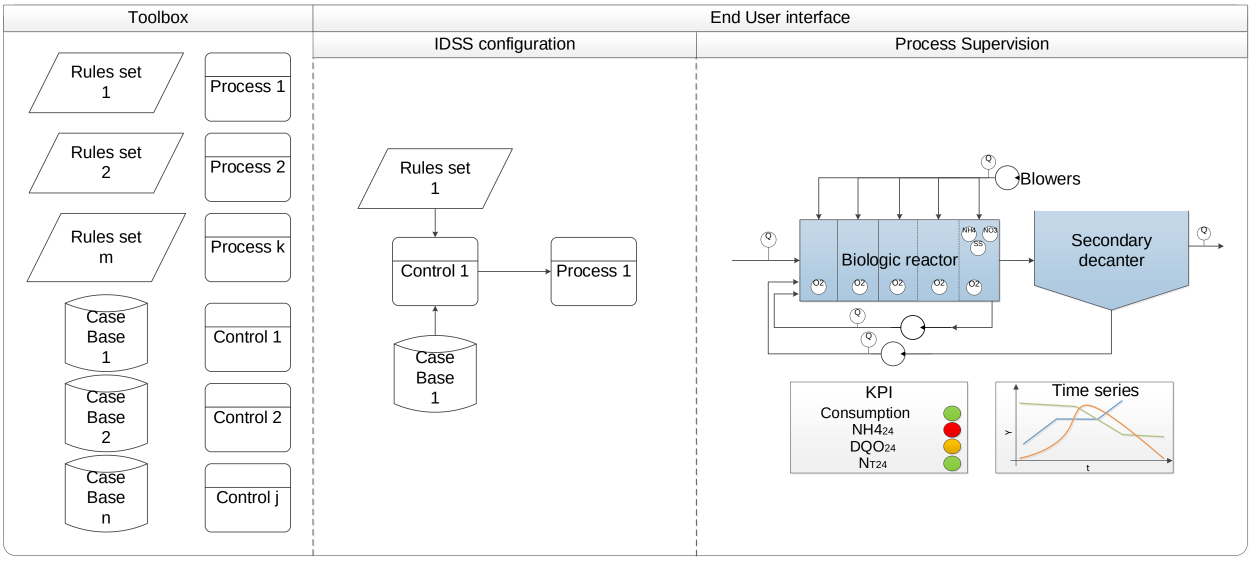


Figure 3.4 Process design layer

3.2.3 Process control flow

The Process Control workflow layer (Layer 3) is the application core, called here IPCS. As it is shown in Figure 3.5, the proposed approach combines Case-Based Reasoning (CBR) and Rule-Based Reasoning (RBR) algorithms, obtaining redundancy in diagnosis and/or set-points generation. CBR and RBR modules inputs are fed with online data gathered from the process and the corresponding models generated in Layer 1. Using both methods and different models obtained with the same data, diagnosis results and set-points obtained can be compared in order to provide a more reliable diagnosis and set-point generation. Hence, this diagnosis and set-point generation redundancy helps on relying on the outcome of the IDSS. Also, human expert knowledge –provided e.g., by the process operator– should be considered not only to generate valid rule models in Layer 1, but also in order to validate the tool outcome and to feed the CB with human-based knowledge. In the next sections, more details about how the user can interact with this process and the importance of the automation of the whole work flow, i.e., from data acquisition to diagnosis, are given

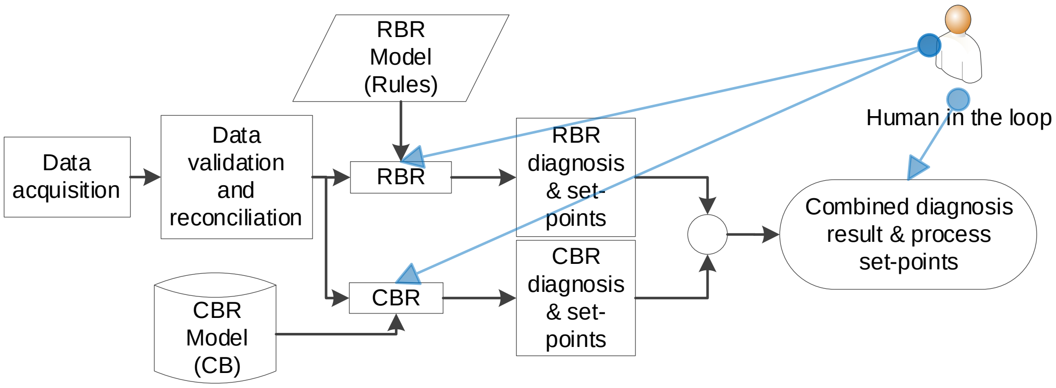

RBR Model (Rules)
Data acquisition
Data validation and reconciliation
RBR
RBR diagnosis & set-points
CBR diagnosis & set-points
CBR
CBR Model (CB)
Human in the loop
Combined diagnosis result & process set-points

Figure 3.5 Process control layer

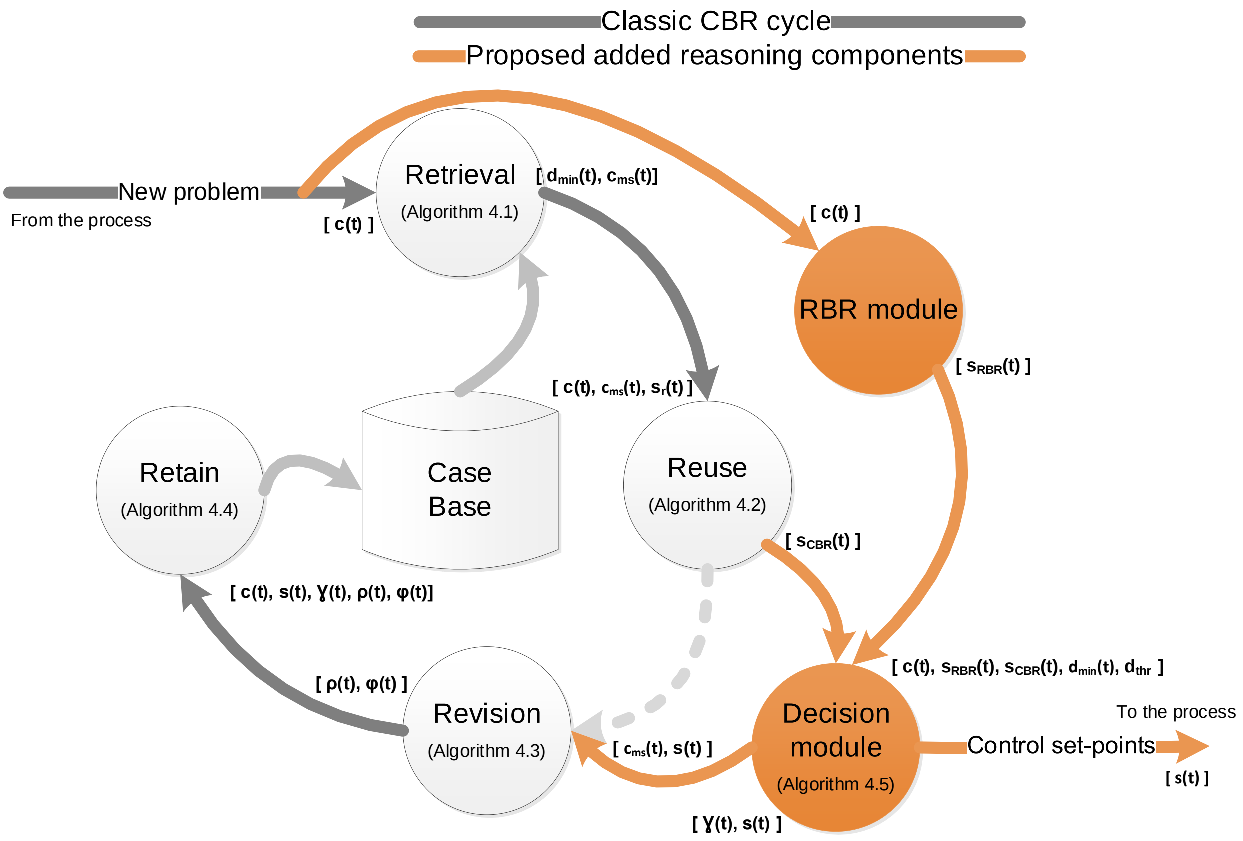
3.3. Conclusions

In **Chapter 3,** a novel hybrid framework based on the interoperation of RBR and CBR techniques for the development of an IDSS, is proposed. The aim of this proposal is to solve a common challenge in the design and implementation of decision support and control systems, which is the ad-hoc design for different installations, as well as the lack of adaptability to dynamic changes on environmental systems. To this end, the presented approach has been designed in a general fashion and integrated in a software tool for the sake of scalability to different types of systems, here WWTPs, but without loss of generality, also to further types of systems beyond the environmental systems domain. The proposed framework allows to differentiate between the core of the application (Layer 3), which consists on the reasoning system presented in Chapter 4, and the models discovered in the Data Science layer (Layer 1). Consequently, the IDSS can be deployed in any installation without modifications. Then, it can be calibrated using valid models for each specific process. All models and other parameters, such as the connection to the process or the KPI, are structured in a PostgreSQL database, which in turn can be reused in any installation with the same configuration.

4. Integrated Reasoning system

As described in Subsection 3.2.3, the proposed IDS methodology is based on the interoperation of RBR and CBR modules to tackle the control and supervision of processes, applied to waste water treatment domain in the context of this thesis (Pascual-Pañach et al., 2019a; 2021). In the next sections, proposed RBR and CBR modules are described, as well as how they are integrated in a unique tool combining both methods in order to obtain a more reliable solution. The CBR component provides the IDS methodology with learning capabilities. Due to the dynamic nature of CBR, the IDSS can increase its competence skills along time, because it can learn from relevant situations experienced, previously solved facing the system supervision, day after day. The RBR module provides a tool to include expert knowledge in the IDSS, which may be used in the situation where CBR cannot provide a reliable solution, or even when historical data is of poor-quality or non-existent in order to initialize an empty case base.

The integration of both modules is shown in Figure 4.1.



$c(t)$: case at time t

$c_{ms}(t)$: most similar case to c(t)

$d_{min}(t)$: distance to the most similar case

$s_r(t)$: Retrieved solution at time t

$s_{RBR}(t)$: solution from RBR module at time t

$s_{CBR}(t)$: solution from CBR module at time t

$s(t)$: solutions at time t

d_{thr}: Distance threshold

$c_{ms}(t)$: Most similar case to c(t)

$\gamma(t)$: RBR solution selected; new case candidate.

$\rho(t)$: Revision fulfilment index

$\varphi(t)$: Fulfilment flag

kpi(t): Key Performance Indicators (KPIs) values vector

Figure 4.1 Interoperation of RBR, CBR and Decision modules scheme

The grey part of the diagram corresponds to the classic CBR cycle, whilst in orange our proposal integrating the RBR module is represented. The Decision module –after rules evaluation and case-based reasoning retrieval phase–, is used to decide which solutions are selected to be applied to the process. Then, at the revision stage, the process KPIs are evaluated for the model used –RBR or CBR– and compared with the defined target values. Finally, in the retain phase, relevant information can be added to the case base. The retrieval, reuse, revision and retain phases of CBR, as well as the RBR Decision modules are described in the next sections.

4.1. RBR approach

The RBR module is designed following the scheme presented in Section 2.5, Figure 2.3. The knowledge base consists of a set of rules to generate the set-points to control the process for which they were designed. Each rule is expressed as in (4-1) as follows:

$$\boldsymbol{If} < condition > \boldsymbol{then} < action > \tag{4-1}$$

The *condition* statement in (4-1) depends on any measured/calculated variable/parameter related with the process that can be modified by the user's tool. It can be a simple statement or a combination of different conditions, by means of logical {AND, OR} operations.

The *action* statement in (4-1) is related to each set-point of the process control. An action can involve setting a set-point to a specific value, or to increase/decrease the current set-point.

The set of rules of the form in (4-1) is designed together with the system manager and experts on the process in a participatory task.

4.2. CBR approach

The CBR module uses a CB obtained from historical operational data of the process. The CB consists of a set of cases and can be represented as in (4-2):

$$CB = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_I \end{bmatrix} \tag{4-2}$$

where I is the number of cases in the CB.

Each case is formatted in a vector form with a set of descriptive features f –sensor measurements or derived variables from these measurements here– and a set of solution parameters s –control set-points, i.e., the desired target values for the essential control parameters, here– as in (4-3):

$$c_i = (f_{i1}, f_{i2}, \dots f_{iN}, s_{i1}, s_{i2}, \dots, s_{iM}); i = 1 \dots I \tag{4-3}$$

where f are the descriptive features, N is the number of features, s are solutions, M is the number of solutions and I is the number of cases in the CB. When new data is read from the process, they are formatted as a new case $c(t)$, following the format in (4-3), so that it can be compared with each case in CB. All features f and solution parameters s are limited within an interval $[f_n^{min}, f_n^{max}]$ and $[s_m^{min}, s_m^{max}]$ respectively, depending on the range of each measurement or control set-point. This comparison is performed in the first stage of the CBR cycle, the retrieval process, which is introduced in Algorithm 4.1. The distance d between the current case at time t $c(t)$ and all cases c_i in the CB is calculated. Then, the case with the lowest distance (d_{min}) to the current one is identified as the most similar case (c_{ms}).

Algorithm 4.1 – Retrieval process

```
function $retrievalFcn(c(t))$
    for $i$ in $1$ to $I$
        $d(i) \leftarrow computeDistances(c(t), c_i)$
    end for
    $(d_{min}, c_{ms}) \leftarrow identMostSimilarCase(d)$
    return $\langle d_{min}, c_{ms} \rangle$
end function
```

To calculate the distance between two cases all features are normalised between 0 and 1, considering the range, i.e., minimum and maximum values, for each variable. This distance can be calculated using different metrics (Núñez et al., 2004). Here, the Euclidean distance has been proposed, mainly for two reasons: a) all the variables considered in the proposed case study are continuous and numeric, b) it is a weighted similarity measure, so less important features may be penalised. Thus, the Euclidean distance between two cases c_a and c_b can be calculated as in (4-4):

$$d(c_a, c_b) = \sqrt{\sum_{n=1}^{N} w_n (f_{an} - f_{bn})^2} \tag{4-4}$$

where N is the number of features in a case, f_{an} and f_{bn} are the normalised values for feature n of each case a and b, respectively, and w_n is the weight of the feature n. By default, w_n is $\frac{1}{N}$ for all n, thereby, all features have the same importance.

Anyway, other similarity measures can be explored in the future, e.g., the one in Serrà and Arcos, 2012, which proposes a new similarity measure based on Minimum Jump Costs (MJC) and compares this approach with the Euclidean and Dynamic Time Warping (DTW) dissimilarity measures. Naturally, applications requiring similarity measures for other types of variables, e.g., qualitative variables, can implement a properly method in the retrieval algorithm.

At the current stage and by default, the most similar case is picked, although other alternatives could be considered, e.g., the k most similar cases (with k being a positive integer).

The second stage –reuse stage in Figure 4.1– is introduced in Algorithm 4.2. In the *reuse* stage, the solution obtained in the *retrieval* stage can be adapted to the new problem requirements. Since the current case may not be exactly the same as the retrieved one (i.e., $d_{min} > 0$), the appropriate solution may not be the same either. Hence, a method can be used to adapt the solution proposed in the retrieved case c_{ms} to solve the situation described in the current case *c(t)*. In general, any of the adaptation methods described in Chapter 2, Subsection 2.3.4, can be implemented and integrated in Algorithm 4.2. Here, null adaptation method (Alterman 1988) is proposed, i.e., no action is performed to the retrieved case and the retrieved solution is used. However, in real systems it may happen that the proposed set-points cannot be applied because of multiple causes: actuators used to reach these set-points may be unavailable or in maintenance tasks, or its operation range limited due to some decision of the plant manager to deal with some unusual situation. Thus, solutions –set-points in the context of the described case study– obtained from CBR retrieval stage can be adapted to these non-ideal situations by considering the interval of limits for each set-point.

Algorithm 4.2 – Reuse process

```
function reuseFcn($c(t), c_{ms}, S_r$)
    for $m$ in $1 : M$
        if $S_{CBR}(t,m) \in S_r(k)$ then //Null adaptation, solution within the current limits.
            $S_{CBR}(t,m) \longleftarrow S_{ms}(m)$ //Solution of most similar case is used.
        else //Null adaptation, but solution out of the current limits.
            $S_{CBR}(t,m) \longleftarrow set - points\ value\ limited\ to\ [s_m^{min}, s_m^{max}]$
        end if
    end for
    return $\langle S_{CBR}(t) \rangle$
end function
```

S_r is the matrix in (4-5) with the range of valid values for each solution, where (s_m^{min}, s_m^{max}) are the minimum and the maximum values allowed for the solution m,

$$S_r = \left((s_1^{min}, s_1^{max}), (s_2^{min}, s_2^{max}), \dots (s_M^{min}, s_M^{max}) \right) \tag{4-5}$$

$S_{CBR}(t, m)$ is the adapted solution obtained in the reuse stage at time step t for the solution variable m, while $S_{ms}(t, m)$ is the solution of the most similar case, i.e., before the reuse process.

The next step is the *revision* stage, which is detailed in Algorithm 4.3. The aim of this stage is to evaluate if the given set of set-points have been successful. Set-points assessment is done evaluating a set of KPIs that have been defined to measure the performance over the time for the most important objectives of the process. In Section 7.3 KPIs for the particular case study presented in Section 7.1 will be described.

Algorithm 4.3 – Revision process

```
function revisionFcn($c_{ms}$, $S(t)$)
    if $t = 0$ then //Initialization of  Utility mesures the first time
        [nuses, okuse, nokuse] = InitializeUtilityMeasures
    endif
    kpi(t) ⟵ CalculateKpiValues
    $\theta(t)$ ⟵ ObtainRevision($kpi_{1..Q}(t), \delta_{1..Q}$)
    if $\theta(t) > \theta_{th}$ then //Fulfilment index is considered correct, no user validation required
        okuse(i) ⟵ okuse(i) + 1 //Increase okuses for the most similar case i.
        $\varphi(t)$ ⟵ 0
    else
        $\varphi(t)$ ⟵ 1 //Solution from case $i$ at time $t$ needs user validation
        list(end+1) = i;
    end if
    When any $\varphi(t)$ from 1 to 0 // The user has validated the solution at time t
        if $\theta(t) == 1$ //User considers that the solution is correct
            okuse(i) ⟵ okuse(i) + 1 //increase okuses for the most similar case i.
            Call retainFcn for time t.
        else
            nokuse(i) ⟵ nokuse(i) + 1 //increase nokuse for the most similar case i.
            //give a short description about the problem.
        end if
    end when
    return $\langle\theta(t), kpi(t), \varphi(t)\rangle$
end function
```

In Algorithm 4.3, KPIs values are calculated in the function *CalculateKpiValues*. Also, other performance measures are considered in order to facilitate the maintenance of the CB: the total number of usages per case (*nuses*), the number of incorrect usages per case (*nokuse)* and the number of correct uses per case

($okuse$). These measures can give valuable information about the CB, e.g., cases usability to identify most common situation or the exceptional ones, or even wrong retained cases if they are never used, or used incorrectly. The revision of the solution at time step *t* is done in the next time step *t+1*. Although sometimes the effects of the used set-points may take some time to be visible, the time between to actuation could be enough for an evaluation of the results.

After KPIs calculation, the fulfilment index $\theta(t)$ is obtained in the function *ObtainRevision* as described in equation (4-6),

$$\theta(t) = \frac{1}{Q} \cdot \sum_{k=1}^{Q} kpi_q(t) \leq \delta_q \tag{4-6}$$

where Q is the number of KPIs, $kpi_q(t)$ is the value of the q^{th} KPI at time step *t* and δ_q is the threshold corresponding to the q^{th} KPI. The δ_q threshold is fixed depending on the nature of the KPI, e.g., effluent quality limits in the waste water treatment directive or a maximum electrical consumption objective. A value of $\theta(t)$ equal to 1 means that all values in *kpi* vector are within the desired limits. On the other hand, a value of $\theta(t)$ below 1 may indicate that recently generated set-points are not best suited to control the process. The lower the value of $\theta(t)$, the greater the number of KPIs out of the desired limits. For example, for a set of 4 KPIs, a value of $\theta(t) = 0.75$ means that three of four KPI values are within the desired limits. The revision function returns a structure with the overall performance index $\theta(t)$ the value for each KPI and the flag $\varphi(t)$. $\varphi(t)$ takes a value of 1 when the KPIs fulfilment index $\theta(t)$ is lower than the threshold θ_{th}, and it is used as an alarm signal for the user when the fulfilment index is not the desired one. Then, the user has to check the solution and decide whether it is considered as a correct solution or not. If not, the user can check the retrieved case and compare it to the current one and decide how to proceed. The most probable options are:

a) the user considers that the proposed set-points are the correct ones, despite one or more KPIs are out of bounds. The deviation in the performance might be caused for a wide range of reasons. In a WWTP, for example, pollution load of the influent waste water over the normal limits, maintenance tasks or failures which limits the capacity of the plant, etc. The user must assess whether the given solution is the best possible one.

b) the user confirms that the proposed set-points are not correct, or at least, they are not the best possible. Again, there are several possible reasons for this: incorrect values in the current or retrieved cases due to a sensor problem, the retrieved case is not similar enough to the current one, or, conversely, they are very similar but represent different situations because of the lack of information. The lack of information problem occurs when the difference between two different situations would be identified by a parameter which is not available or unmeasured. Without the value of that parameter, these two situations cannot be distinguished.

It can also happen that automatically stored cases, i.e., those that do not require the user, may be incorrect. Thus, it is necessary to provide tools to the user to add new cases and remove or modify the existing ones in the CB, depending on the result of the revision stage.

An important remark is that the effects of a certain actuation depend on the process dynamics and on different boundary conditions like, for instance, in the case of a WWTP, the influent load. This means that the cause behind the deviation of one or more KPIs may not be due to the last and most recent actuation, but because of a set of successive incorrect actuations or even external causes. Thus, the revision result has to be considered as a guide for an operator to find the cause of the problem. When the revision process is not passed, i.e., $\theta(t) < \theta_{th}$, the expert can verify whether the last actuations –e.g., the last few hours– are correct or not. At this point, there are several possibilities: all actuations are considered as correct by the operator, so no apparent cause in the decisions is found and the revision result $\theta(t)$ can be updated to 1; the operator finds the cause of the out of limits KPI, then the CB should be checked and detected incorrect cases can be corrected. In the first situation, the non-fulfilment of the KPIs can be caused by an external and probably exceptional situation, e.g., contaminant discharge over allowed limits, so the expert can modify some parameters of the process to be adapted to that environmental situation, or just be aware of that and assume that the best actuation has been applied to the system. In the second one, also the rules can be checked because depending on the Decision module describe in the next section, set-points can be generated by CBR or RBR modules.

The last stage in the CBR cycle is the *retain stage* (Algorithm 4.4).

Algorithm 4.4 – Retain process

```
function retainFcn($c_0(t), S(t), \gamma(t), \theta(t), \varphi(t)$)
    if $\gamma(t) = 1$ then //RBR solution is used, then $c_0(t)$ is candidate to be retained.
        if $\theta(t) \geq \theta_{th}$ then // Solution at time t is validated → positive revision
            $CB(l+1) \leftarrow c_0(t)$ ;
            $l \leftarrow l+1$;
        elseif $\theta(t) < \theta_{th}$ and $\varphi(t) = 1$ then
            Candidate case $c_0(t)$ cannot be retained yet. User validation required.
        else
            $c_0(t)$ is not retained. User validation is not passed.
        end if
    elseif $\gamma(t) \neq 1$ then
        $c_0(t)$ is not a candidate → Not retained.
    end if
    return $\langle CB \rangle$
end function
```

In the *retain stage* relevant situations that are not represented in the CB can be learned -aggregated to the CB- to be used in the future, increasing the competence of the IDSS along time. Candidate cases to be learned are identified in the Decision module (Algorithm 4.5): if the Decision module outcome –RBR vs. CBR– is RBR, –CBR module cannot solve the environmental problem–, the decision flag $\gamma(t)$ is set to 1 and the new case will be considered as a candidate in the Retain stage. It is important to learn from well-solved problems. The final decision depends on the revision result $\theta(t)$. If $\theta(t)$ is over θ_{th}, then the current case $c_0(t)$ is added to the CB; otherwise, $c_0(t)$ is rejected. When the revision is pending (flag $\varphi(t) = 1$) because expert's validation is required, the retain process have to be postponed until the user validation. Additional details on the Decision module are given in Section 4.3.

Finally, special attention should be given to the cases *retention* to avoid an information overload. The information contained in the CB should represent all the possible situations involving the process, while storing the minimum number of cases (i.e., rows in the CB). A large CB involves more resources in terms of physical memory to allocate it and computation time in the retrieval stage. What it is expected is a behaviour similar to the one represented in Figure 4.2. At CBR initialization, the number of new retained cases can be high, but it should decrease over time. Only when a new situation occurs, the number of retained cases will increase until that new situation is learned. Thus, under normal conditions, the case base should not be overloaded. If that happens, it may be an indication that something is wrong with data the criteria for case learning. The next section explains how the distance threshold can be set.

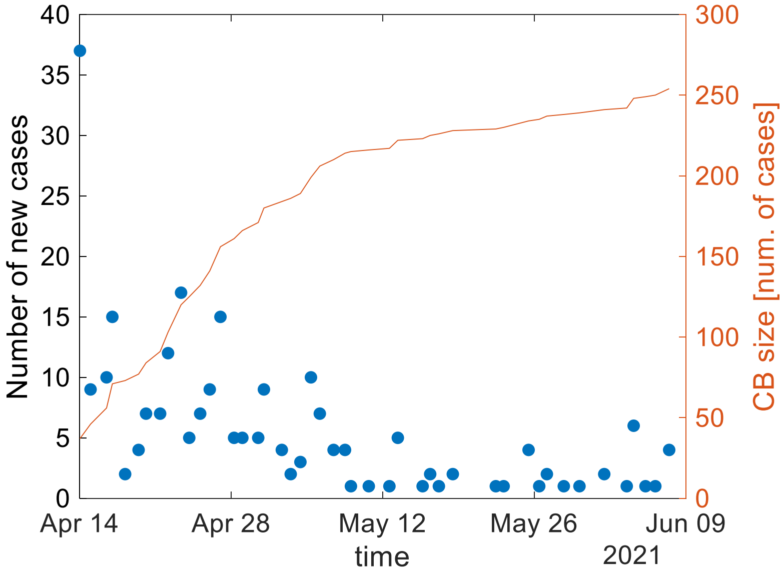


Figure 4.2 Number of retained cases over time.

However, some measures can be considered to keep the CB with the amount of necessary and reliable information. *Performance measures* in Algorithm 4.3 can be useful for this purpose, e.g., incorrect used cases during a certain period can be deleted.

4.3. Combination of RBR and CBR approaches

The RBR and CBR modules are interoperating through the Decision module. The CBR approach provides more specific knowledge and learning capacity in comparison to the RBR approach. For this reason, the solution proposed by the CBR module is more reliable when the computed dissimilarity measure between the current case and the most similar one in the CB is below a threshold. In spite of this decision can be automated, the critical nature of this application makes it essential to provide tools to involve the participation of the user in different decisions along the management of the process, like in the *revision* and *retain* stages of the CBR cycle or the Decision module. At each time step the solutions (set-points to control the process in this case) are generated. The dissimilarity measure (d_{min}) of the retrieved case is used to determine the solution reliability. Here, the dissimilarity (d_{min}) of the current case to the retrieved one is compared to d_{thr} calculated in (4-9) to decide on the solution to be used. With a distance smaller than d_{thr}, it is assumed that, first, the situation described by the current episode does not provide additional information to be added in the CB – the case is not a candidate - , and second, that the given set-points can be used, after the reuse step, to solve the current problem because the current situation is similar enough to the one occurred in the past.

The distance threshold can be statistically obtained, but also fixed and modified by the user depending on the confidence on the CBR module. In the first case, assuming a case base CB, a set of P experiences or cases that can be solved with the CB and taking the minimum distance of each case to the most similar one, the distance threshold d_{thr} is obtained as in (4-9). First, the average μ_d of all minimum distances is calculated in (4-7),

$$\mu_d = \frac{1}{P}\sum_{p=1}^{P} d(c_p, c_{ms}) \tag{4-7}$$

where $d(c_p, c_{ms})$ is the distance of the case c_p to its most similar case in the CB, i.e., the minimum distance, and P is the total number of solved cases. This average distance μ_d can be recalculated as the CB size increases. Then, the standard deviation of all minimum distances is calculated in (4-8):

$$\sigma_d = \sqrt{\frac{1}{P}\sum_{p=1}^{P}\left(d_p - \mu_d\right)^2} \tag{4-8}$$

where d_p is $d(c_p, c_{ms})$. Finally, the distance threshold is calculated in (4-9) and used as shown in Algorithm 4.5 to determine which solution has to be used, the one provided by the RBR model ($S_{RBR}(t)$) or the one from the CBR module $S_{CBR}(t)$).

$$d_{thr} = \mu_d + 3 \cdot \sigma_d \tag{4-9}$$

Algorithm 4.5 – Decision module

```
function $decision(c(t), S_{RBR}(t), S_{CBR}(t), d_{min}, d_{thr})$
    if $d_{min} \leq d_{thr}$ then
        $CBR\ is\ reliable \rightarrow S(t) \longleftarrow S_{CBR}(t)$
        $\gamma(t) \longleftarrow 0$
    else
        $S(t) \longleftarrow S_{RBR}(t)$
        $\gamma(t) \longleftarrow 1$
    end if
    return $\langle \gamma(t), S(t) \rangle$
end function
```

When the distance to the retrieved case is below d_{thr}, it is assumed that the current situation is enough similar to the retrieved one and, consequently, the solution given by the CBR module can be used. On the other hand, a distance value over d_{thr} means that the current situation is not similar enough to any stored case in the CB. Thus, the user is warned and the problem is solved using the rules from the RBR module. The threshold d_{thr} can be changed by the user, increasing or decreasing its value depending on the confidence on the CBR module. Using the extreme values, it is possible to cancel one of both modules. With $d_{thr} = 0$, the RBR module is always used, while with $d_{thr} = 1$, the use of the CBR module is forced.

4.4. Conclusions

In **Chapter 4**, the reasoning system, which is the core of the IDSS proposed in Chapter 3, is presented. RBR and CBR modules are first described separately, and then, combined to increase the reliability of the whole IDSS. The CBR module provides a more specific knowledge and learning capacity in comparison with the RBR module. The RBR module enables the user to include expert rules which can complement the CB with more generic knowledge that can be used when the CBR could not reliably solve the set-points generation problem. The use of explainable methods such as RBR and CBR facilitate the development of this framework, allowing the characterization of each installation through the expert knowledge encoded in a set of rules or using available data in the form of a CB.

5. Temporal CBR Approach

5.1. Temporal Case-Based Reasoning Terminology

In dynamic domains, a state at a particular time depends on past states. It means that the identification of situations occurring in a process or the suggestion of set-points to control that process may not be very accurate when using single cases (Sanchez-Marrè et al., 2005). In this former research work, the authors suggest that cases can be grouped in sequences of cases making up episodes. Episodes consist of a variable-length sequence of consecutive cases, depending on the diagnosis of each individual case, i.e., same diagnosis means same episode. They perform a preliminary evaluation of the method in the waste water treatment domain using available offline data, obtaining an improved performance with respect to the basic CBR. In view of the potential of this research line and its applicability to the waste water treatment domain, in this chapter a new approach using fixed-length episodes is proposed (Pascual-Pañach et al., 2022c).

The assumption here is that using more than one case will enable a better identification of the current situation. For example, at a particular time step, and considering the whole feature set, a high load situation is identified. But to decide about which actions should be taken, it would be interesting to know if the high load episode is getting worse or better. If the situation is getting better it would mean that the current actions are working as expected, otherwise some actions should be changed. Considering temporal information to describe the current situation can be done with both, fixed or variable-length episodes. In this work it is proposed to find a fixed-length to maximize the performance of the IDSS, i.e., improve the accuracy of the retrieval process. The advantage over the variable-length approach is that it is not dependent on the diagnosis. As noted, each episode with a particular diagnosis may have a different duration, since this diagnosis can be shorter or longer in time.

In the classical CBR approach, the CB consists of a set of cases which are not necessarily consecutive ((4-2) in Section 4.2) –they are single static case captions taken at a particular time. Alternatively, in a temporal domain, cases are not isolated events, but sequences of time-related cases forming episodes describing a particular situation, e.g., in the waste water treatment domain, a contamination event. Thus, the Episodes Base (EB) consists of a set of I episodes e_i represented as in (5-1):

$$\text{EB} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_I \end{bmatrix} \tag{5-1}$$

An episode e_i can be defined as a set of J consecutive cases c_j, where c_1 is the initial case of the episode and c_J is the last case of the episode. In addition, an episode description is formed by the structure in (5-2), including other components:

$$\begin{aligned}&\text{Episode } e_i\\&:\text{id: } 1..I\\&:\text{initial case time: } t_0\\&:\text{final case time: } t_f\\&:\text{episode length: } J\\&:\text{episode diagnosis: } d_1, d_2, \dots, d_l\end{aligned} \qquad (5\text{-}2)$$

where t_0 and t_f are the time stamps corresponding to the beginning and the end of the episode, J is the number of cases between t_0 and t_f, i.e., the episode length, d_1 *to* d_l is a set of l labels describing the episode, i.e., the episode diagnosis.

Episode diagnosis, which is mandatory for *variable-length approach*, describes the episode with a set of predefined labels. Several labels can be considered since several different situations or problems can happen in any environmental process at the same time. The diagnosis is obtained for each new case, but as mentioned above, only *variable-length episodes* are composed of cases with the same diagnosis. In the variable-length approach, the diagnosis of cases determines when an episode is finished. This means that cases with the same diagnosis, i.e., a particular status of the process, will be part of the same episode. Thus, this common label of consecutive cases is the one that can be assigned to the episode. When a change in the process results in a change in the diagnosis of a case, then that case will start a new episode. On the other hand, in the *fixed-length episode approach* proposed here, cases forming the episode do not necessarily have the same diagnosis (although usually they have the same). They could have different diagnoses inside the episode, especially when the episode length is a large number. Diagnosis labels can be generated using diagnosis rules provided by the expert in the process or by the comparison with case prototypes for different situations.

Assuming the definition of a case in (4-2) Chapter 4, and from a temporal point of view, an episode consists of a concatenation of consecutive cases as expressed in (5-3).

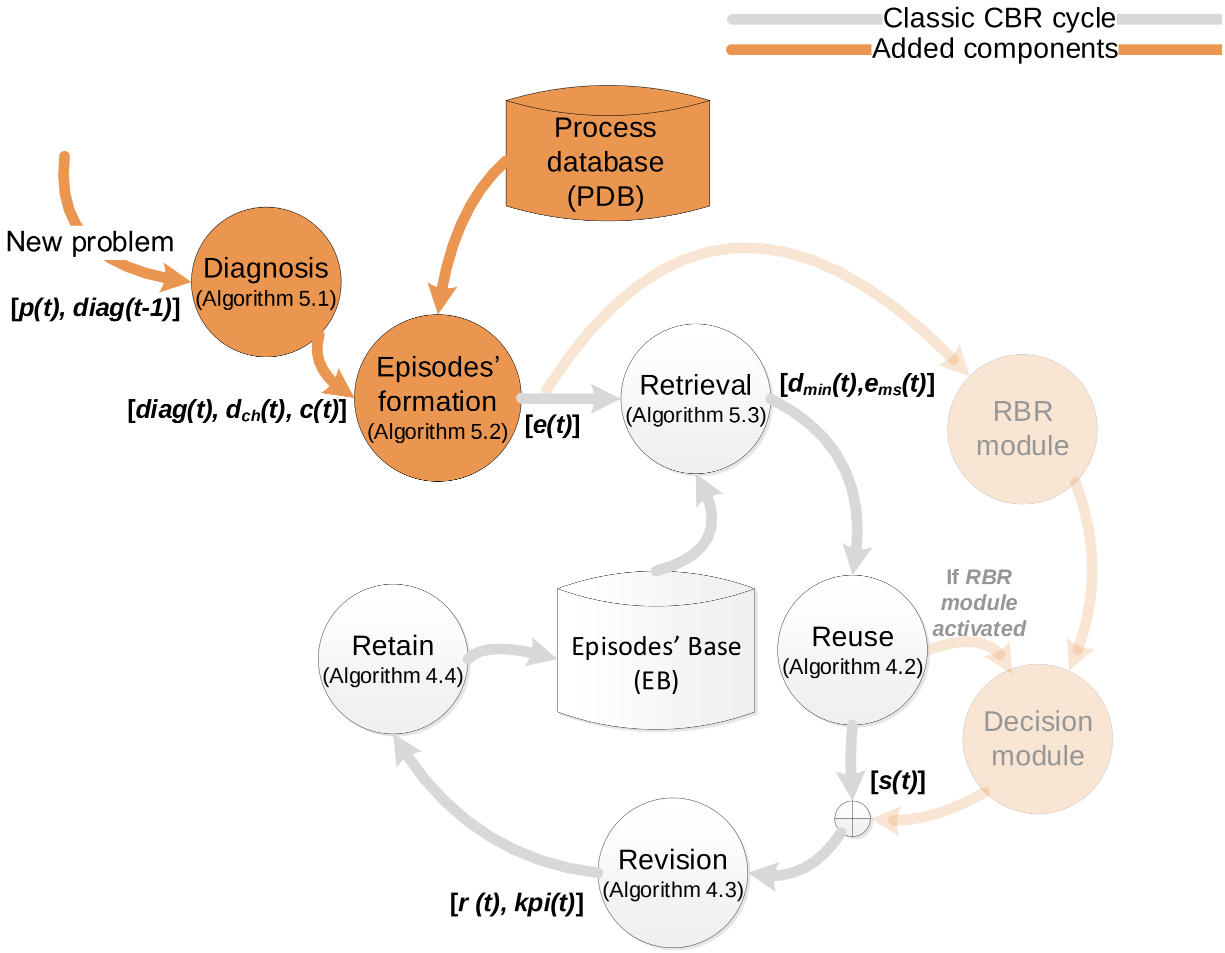
$$e_i = (c_t, c_{t+1}, c_{t+2} \dots c_{t+J-1}) \qquad (5\text{-}3)$$

where t is the current initial time index for the episode e_i. The EB can be obtained using historical data from the process.

5.2. Temporal Reasoning Approach

As mentioned in Section 5.1, the temporal CBR approach proposed here is based on *fixed-length episodes*, i.e., all episodes consist of the same number of consecutive cases. Despite this approach, the reasoning cycle is designed in a general fashion that enable the use of both *fixed-length* and *variable-length episodes*.

The proposed modified CBR cycle is shown in Figure 5.1. The proposal in Figure 4.1 from Chapter 4 have been taken as a starting point. But here the focus is on the previous stages of the Retrieval one (in orange). Components added and explained in Chapter 4 are represented here in a lighter orange.



$p(t)$: problem at time t
$diag(t-1)$: diagnosis at time $t-1$
$diag(t)$: diagnosis at time t
$d_{ch}(t)$: diagnosis change flag

$c(t)$: case at time t
$e(t)$: episode at time t
$d_{min}(t)$: distance to the most similar episode
$e_{ms}(t)$: most similar episode to $e(t)$

$s(t)$: solutions at time t
$\rho(t)$: KPIs fulfillment index
$kpi(t)$: Key Performance Indicators (KPIs) values vector

Figure 5.1 Proposed reasoning scheme extending the classic CBR cycle

These previous steps are the Diagnosis stage (Algorithm 5.1) and Episodes' formation stage (Algorithm 5.2). Both stages are executed each time a new problem $p(t)$ is identified. For the episode generation, the data collected from the process in the process database (PDB) is used. Unlike the classical CBR approach, where the case is formed using data from the current time t, in the temporal approach data from previous instants is required. The RBR and Decision modules are described in Chapter 4, but they are kept in the scheme because they are independent of the proposed approach. The CBR part is described emphasizing the differences with the explanations in previous chapters. At this point, it is necessary to highlight that the RBR module and the CBR system are independent, – i.e., they have to solve the same problem, giving the same outputs, but the RBR module does not depend on the CB or EB. Therefore, the proposed temporal approach is applied only to CBR; the RBR module can address the problem from a static approach, considering only values from the current time step.

Algorithm 5.1 – Diagnosis (labelling)

```
function $diagnosisFunction(p(t), diag(t-1))$
    $c(t) \leftarrow problem2case(p(t))$
    $diag(t) \leftarrow evalDiagnosisRules(c(t))$
    if $variable\ length$
        if $diag(t) = diag(t-1)$
            $d_{ch} \leftarrow 0$
        else
            $d_{ch} \leftarrow 1$
        end if
    end if
    return $\langle diag(t), d_{ch}, c(t) \rangle$
end function
```

The first stage from the cycle in Figure 5.1 is the Case Diagnosis. The first step within the Diagnosis algorithm is to build the case from the problem description. The function *problem2case* in Algorithm 5.1 takes the available data gathered from the process (problem $p(t)$) and generates the case $c(t)$ formatted as in (4-3), considering all values defined as features and concatenating them in the correct order. Then, the function $evalDiagnosisRules$ evaluates the case, which is assigned a set of diagnosis labels, i.e., diagnosis of the process, returning $diag(t)$. There are many formalisms in Artificial Intelligence (AI) to represent the knowledge available in a concrete domain, such as ontologies, semantic networks, frame systems, inference rules, etc. All them have advantages and drawbacks. Structured representations allow to model complex relations among different elements of the domain, but their inference processes are slower than using other non-structured formalisms. In our case, as the expert knowledge needed for the diagnosis process is rather simple, without complex relations and dependencies among domain elements, inference process is fast, and also experts feels more comfortable describing their knowledge in rules' format, so we selected this formalism. However, if more complex knowledge about the domain should be modelled, a structured representation formalism would be the best choice. We indeed think that a rule-based reasoning mechanism using inference rules is an appropriate and convenient formalism to model the available expert knowledge. In complex systems, considering experts knowledge –both general knowledge or specific knowledge of a particular process– is paramount to deploy a valid and efficient IDSS for its operation. Here, a set of expert rules expressed with the structure in equation (4-1) from Section 4.1 is used, but instead of an action, the consequents is a diagnosis label (5-4).

$$\begin{gathered}
\boldsymbol{If} < condition1 > \boldsymbol{then} < diagnosis_1 > \\
\boldsymbol{else\ if} < condition2 > \boldsymbol{then} < diagnosis_2 > \\
\boldsymbol{else\ if} < condition3 > \boldsymbol{then} < diagnosis_3 > \\
\dots \\
\boldsymbol{else\ if} < conditionN > \boldsymbol{then} < diagnosis_N > \boldsymbol{end\ if}
\end{gathered} \tag{5-4}$$

Each condition statement can consist of different sub-condition statements linked together using logical operators. Moreover, condition statements can use any feature variable forming the case, i.e., any f or s in (4-3)(4-2), or other available information even though it is not included as a feature, e.g., sensors or actuators malfunction. After the diagnosis step and in the case of *variable-length approach*, the current obtained diagnosis $diag(t)$ is compared with the diagnosis of the previous case, $diag(t-1)$, to determine if a new episode should be initialized. This information is assigned to d_{ch} variable, i.e., new episode when $d_{ch} = 0$, which will be used in the following steps.

After the diagnosis step, the episodes' formation step is executed (Algorithm 5.2), returning the current episode $e(t)$.

Algorithm 5.2 – Episode formation

```
function $formEpisode(c(t), diag(t), t, d_{ch}, J, PDB)$
    if $fixed\ length$
        $e(t) \leftarrow createEpisode(c(t), cstruct, t, J, PDB)$
    else if $d_{ch} = 0$ //variable-length
        $c(t)$ added to the current episode (episode update)
    else
        Current episode finished. $c(t)$ to a new episode (episode initialization).
    end if
return $e(t)$
end function
```

In the *fixed-length approach*, the *createEpisode* function uses a fixed number of cases J to create the new episode $e(t)$, which is made up of the current case $c(t)$ at time step t, and the $J-1$ previous cases, i.e. from $t-(J+1)$ to t. This function can retrieve the values for the features and solutions in $cstruct$ from the process database *PDB* to build first, each case as in Algorithm 2, and then, the episode according Equation (4). Although this work is focused on *fixed-length episodes*, in Algorithm 2 is detailed that the diagnosis obtained in Algorithm 1 should be used for episodes creation in the case of *variable-length episodes*. Specifically, the formation of new *variable-length episodes* will depend on the diagnosis changes between consecutive cases (d_{ch}). Thus, in the *variable-length approach* episodes can be updated or initialized depending on case diagnosis.

With the current episode created, the retrieval step is executed. This process is detailed in Algorithm 5.3. The retrieval process is the same described in Algorithm 4.1 in Section 4.2, but with episodes instead of cases.

Algorithm 5.3 – Retrieval process

```
function $retrievalFunction(e(t))$
    for $i$ in 1 to $I$
        $d(i) \leftarrow calculateDistance(e(t), e_i)$
    end for
    $(d_{min}, e_{ms}) \leftarrow identifyMostSimularEpisode(d)$
    return $\langle d_{min}, e_{ms} \rangle$
end function
```

In the retrieval step the current episode $e(t)$ is compared with all stored episodes in the EB. Since *fixed-length* episodes are considered, it is assumed that the length of the current episode $e(t)$ and the length of any episode e_i in the EB is known and constant. The distance between the current episode $e(t)$ and all the episodes in EB –distances vector d of length I episodes –, is computed in *computeDistances* function. Here, as in the approach described in Chapter 4, the Euclidean distance is considered. The same reasons are valid and applicable in the temporal approach, although other similarity measures could be integrated depending on the problem domain. Moreover, considering the *fixed-length* approach, i.e., compared time series have the same length, the distance between episodes can be calculated in a similar way to the one for cases. It is calculated aggregating the Euclidean distance between cases at the same time position from compared episodes, as in Equation (5-5):

$$d(\mathrm{e}_a, e_b) = \frac{1}{J}\sqrt{\sum_{\substack{1<j<J \\ 1\leq n \leq N}} w_n\left(f_{aj,n} - f_{bj,n}\right)^2} \tag{5-5}$$

where e_a and e_b are two different episodes, $f_{aj,n}$ and $f_{bj,n}$ are the values of the feature n in the case j from episodes e_a and e_b, N is the number of features of each case forming an episode, J is the number of cases of an episode, i.e., the episode length, and w_n is the weight of the feature n. By default, and assuming that all features have the same importance, $w_n = \frac{1}{N}$.

After the distance is evaluated for all episodes, the *identMostSimilarEpisode* function finds the episode at the smallest distance d_{min}, i.e., the most similar episode e_{ms}. The minimum value in vector d will identify which episode in EB is the most similar to $e(t)$. Figure 5.2 illustrates the differences in the retrieval process for a CBR approach (described in Chapter 4) versus a TCBR approach.

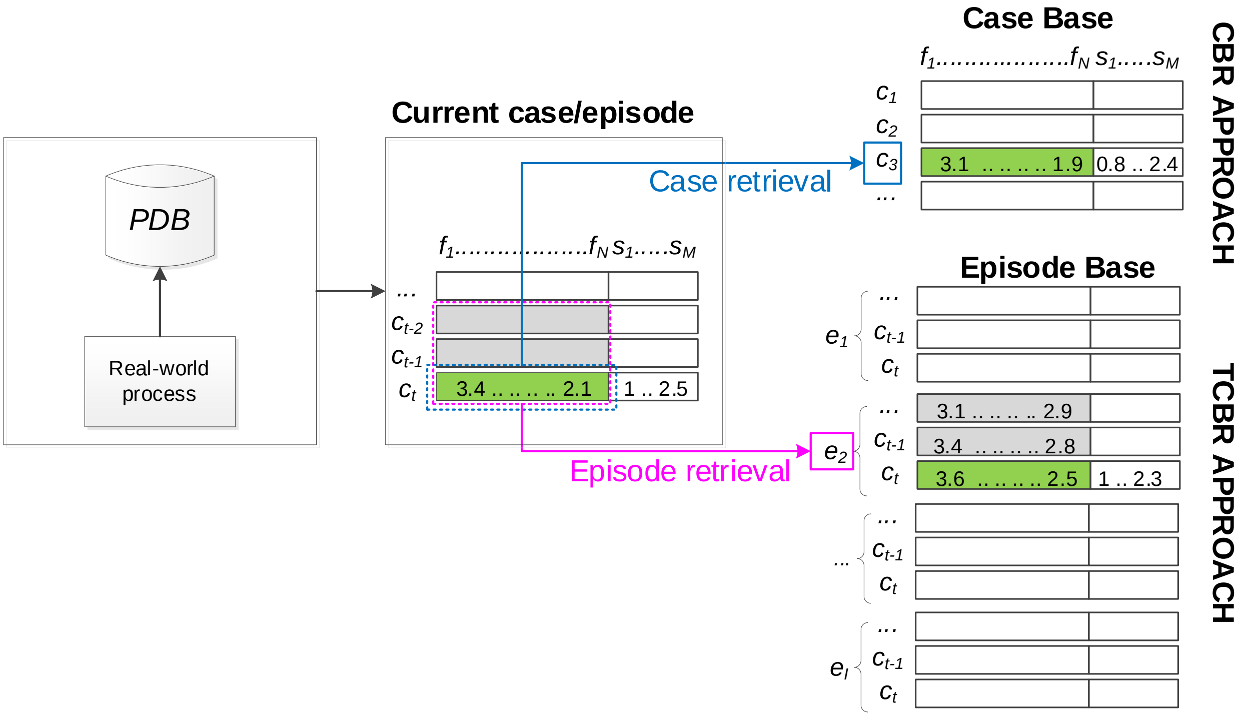


Figure 5.2 CBR vs TCBR problem approaches

The next steps are the *reuse* and the *revision* stages. In the *reuse* process, the solutions s corresponding to the last case of the retrieved episode can be considered and adapted to the current problem. Assuming that the solution parameters corresponding to the most similar episode e_{ms} are s_{J1}^{ms} to s_{JM}^{ms} , the reuse process is performed as described in Algorithm 4.2 from Section 4.2. Note that the solution is the one from the most recent case in the most similar episode, i.e., the case in the position J. Solutions from previous cases are not considered. The method is also null adaptation, but, for safety reasons, it is checked that the values are within the minimum s_m^{min} and the maximum s_m^{max} limits, which can respond to operational needs.

The *revision* phase is based on a set of KPIs and expert rules to evaluate whether the solutions given are correct or not. This process is performed following the same steps in algorithm 4.3 from Section 4.2. Likewise, the *retain* stage can be described as in algorithm 4.4 in Section 4.2. There is no difference in *revision* and *retain* stages from the point of view of how knowledge is represented, in cases or episodes.

5.3. Conclusions

In **Chapter 5**, a temporal version of the reasoning system described in Chapter 4 is proposed. Based on the research in Sànchez-Marrè et al., 2005, a temporal CBR approach is described. To include the temporality in the classic CBR solution, the description of the current status will be more accurate if the dynamics of the cases is considered —i.e. not only a snapshot of the present situation, but also in the preceding ones—. Thus, this modification of the classic CBR approach aims to better identify the process situation. Here, the use of fixed-length episodes is proposed as an alternative to the variable-length approach described in Sànchez-Marrè et al., 2005. The main advantage of this solution is to avoid the dependence on the individual diagnosis of each case. Indeed, the aim of this approach is to improve the accuracy of the retrieval phase, so the idea is to find a fixed-length to maximize the performance of the CBR system.

6. Data Validation and Imputation

Intelligent Decision Support Systems (IDSSs) integrate different Artificial Intelligence (AI) techniques with the aim of taking or supporting human-like decisions. To this end, these techniques are based on the available data from the target process. This implies that invalid or missing data could trigger incorrect decisions and therefore, undesirable situations in the supervised process. This is even more important in environmental systems, which incorrect malfunction could jeopardise related ecosystems. In data-driven applications, such as IDSS, data quality is a fundamental problem that should be addressed (De Mulder et al., 2018), as its performance largely depends on data quality. In real applications data problems are common, i.e., missing or out of range values, abnormal behaviours due to faulty sensors, etc. In (Eliades and Polycarpou, 2002) is pointed out that fault diagnosis and security in water systems are key challenges that will become even more crucial in the years ahead. Thus, it is worth noting that data quality is paramount to ensure a good performance of the proposed methodology in Chapters 4 and 5.

Some recent examples of machine learning methods used for data imputation or to deal with missing values are described in the following works. In Herrera-Vega et al., 2018, the authors propose the use of Bayesian networks to detect and reconstruct not only outliers but also incongruent values in time series. Quesada et al., 2021 models time-series using dynamic Gaussian Bayesian networks and compares its performance with Recurrent Neural Networks. In Ngouna et al., 2020 a data-driven framework for diagnosing causes of water quality contamination is presented. In the latter work, datasets with a high rate of missing values are used. The missing values imputation is done using Chained Equations (MICE) and Support Vector Regression (SVR). In Cheng et al., 2019 an imputation method based on a *k* nearest neighbours algorithm is proposed and applied to a financial prediction problem. Qi et al., 2021 proposes also the use of a reliable k nearest neighbours (RKNN) algorithm applied to incomplete interval-valued data. In Flores et al., 2019, a case-based reasoning approach for offline medium-gaps (from three to ten missing values) imputation is proposed and applied to meteorological time series. Athanasiadis and Mitkas 2004, addresses uncertainty and other data quality issues through the use of data mining techniques. In Caiafa et al., 2021 some novel contributions on machine learning methods with low-quality or imperfect datasets are presented. In Cugueró-Escofet et al., 2016, the combination of spatial models and time-series models to validate and reconstruct invalid or missing data is proposed. Martí-Sarri et al., 2019 propose a novel method based on Tensor-Decomposition to complete the data lost by a SCADA system in case of long bursts

Other works propose the ensemble of different models with the aim of improving the performance of using single models' outputs (Dietterich, 2000). In Kokkinos et al. 2021 an ensemble composed of different methods including Adaptive Neuro Fuzzy Inference Systems (ANFIS), Long Short-Term Memory (LSTM) recurrent neural networks and Extreme Learning Machines (ELM) is proposed to predict traffic-induced pollutants concentrations. Saad et al., 2020 point out the hurdle of missing values when using time series datasets for prediction or forecasting, as well as the inefficiency of conventional imputation methods, like averaging or filling with the last reliable value. They evaluate the use of different deep-learning and machine learning methods, as well as some ensemble methods, for four different types of time series: trend, seasonal, combined and random. Oehmcke et al., 2016 propose an ensemble by combining kNN and

Dynamic Time Warping (DTW) algorithms for the imputation of intervals of missing values in time series. The proposed method is tested using different datasets from the UCR Time Series archive (Dau et al., 2019). Some of these methods are applied to incomplete data-sets with the aim of carrying out a subsequent reliable analysis of these data and are not evaluated for online imputation. Some recent works propose methods for online imputation applied to the waste water treatment domain (Han et al., 2023a, b, Ba-Alawi et al., 2023). The use of NN models is quite common due to its robustness and capability to extract the underlying knowledge and dealing with nonlinear data to solve complex problems. In Han et al., 2023a a solution for recovering random missing data in wastewater treatment processes is presented. The solution is based on the combination of online learning and autoencoder technics, and incorporating a synapse-weighted approach. This proposal demonstrates promising results reconstructing missing data. In Han et al.,2023b a novel filter transfer learning algorithm for imputing missing data in the context of wastewater treatment processes is proposed. Proposed algorithm leverages transfer learning, which involves transferring knowledge from a source domain with abundant data to a target domain with missing data. By incorporating filter-based feature selection and utilizing a domain adaptation strategy, the algorithm effectively imputes missing data in the wastewater treatment process. Ba-Alawi et al., 2023 propose a multisensor fusion approach to impute missing data using an attention recurrent residual CNN (R2AU-Net). Besides, the contribution of sensors on the MBR performances is assessed using explainable AI techniques. In this thesis, the use of explainable techniques such as CBR

Here, the advantage of explainability and learning capabilities of CBR-based methods are proposed in order to impute online data. In the following sections the data validation and imputation methodology for time-series is described. The developed Data Validation and Imputation module is integrated in the IDSS tool described in the following chapters. This integration is described in Section 6.1. Then, in Section 6.2 the problem to be solved is described, detailing parameters such as fault types or length. In Section 6.3 the validation stage is described. Finally, the imputation stage is detailed in Section 6.4.

6.1. Integration of the validation and imputation module

The integration of the data validation and imputation module in the CBR cycle is depicted in Figure 6.1 (Pascual-Pañach et al., 2022a; 2022b). When a new problem is gathered from the process and converted to a case, it is important to validate that all sensor measurements are available and correct. Otherwise, the solution obtained after the execution of the reasoning system may not be the expected one. Thus, it is important to include a preliminary validation and imputation step before the retrieval stage. If all the values forming the case $c(t)$ are available and considered correct in the Data Validation stage, the reasoning process is executed as usual, following the four stages in the CBR cycle in order to determine the required set-points for the process.

If one or more values forming a case $c(t)$ are missing, the imputation stage is executed. Here, as with any imputation method, it should be defined under what conditions the proposed method and models used will be reliable to impute missing values. Imagine a case $c(t)$ consisting of six features, five of them missing or including incorrect values. It is likely that in such situation any imputation method could provide reliable new values. Further details about this casuistic are given in Section 6.2.

The imputation stage is based on a CBR approach and consists of two different stages. First, the retrieval stage aims to find similar cases or episodes to the current one. Then, in the reuse stage, retrieved cases are used provide an estimation of the invalid or missing data. Finally, the reasoning process can run. At this point, it is possible to choose whether to repeat the retrieval stage with the new case $\hat{c}(t)$ or to jump to the reuse stage considering the cases retrieved in the validation and imputation stage in order to provide the set-points to control the process. Thereafter, the next steps are executed following the corresponding algorithms described in Chapter 5.

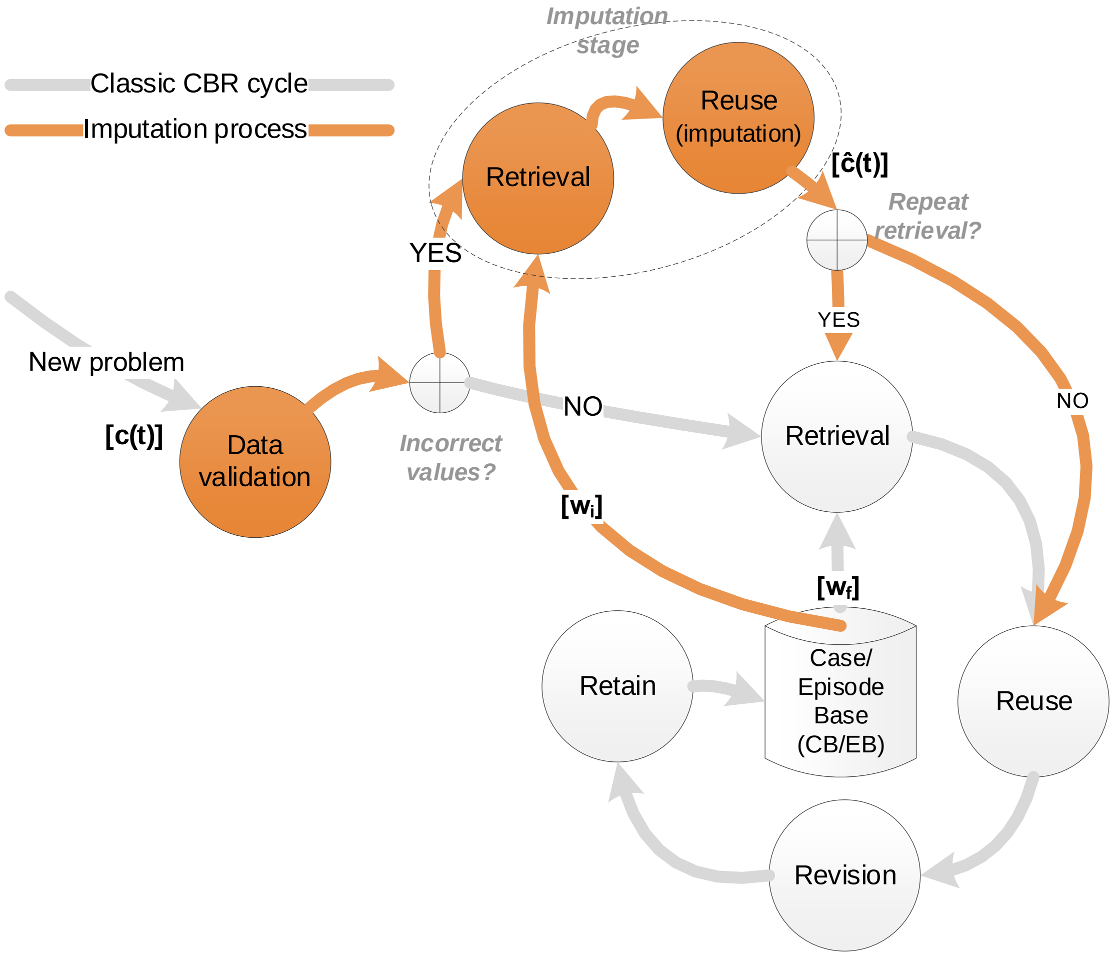


Figure 6.1 Integration of the CBR-based imputation module in the classic CBR cycle

6.2. Problem description

The general context architecture for the IDSS is presented in Section 3.2, while the data acquisition architecture for the case study is described in the following chapter, in Section 7.1. Typically, data obtained from sensors are stored in the database despite its quality, i.e., incorrect values are also stored in the database and used as correct values.

Most usual quality problems can be classified as:

- Communication problems: in this case, the value is not received, so the last available one is used.
- Outlier values: outlier values in a time-series are those that differs significantly or which are not consistent with other observations. Within this context, outlier values are not only out-of-range values, e.g., a value of 12.4 in a sensor with a range in the interval [0 10], but also those that differs from expected values. For example, considering a time series with values ranging from zero to five, being 0.35 the greatest difference between two consecutive samples. Thus, if at time t_0 the value equals 3.4 and at time t_1 the value equals 4.6, this value should be considered an outlier because the increase compared with the previous sample is 1.2, greater than 0.35. In short, outlier values in a particular time-series will be those that do not satisfy the set of conditions defined to describe a normal behaviour of that time-series.
- Incipient faults: As opposed to abrupt faults, incipient faults develop slowly in time, usually due to uncalibrated sensors or sensors degradation. A calibration problem occurs when the sensor is not properly adjusted or calibrated according to a ground truth value. In that case, received information will not be reliable. The detection of a calibration problem is not as simple as the detection of an isolated outlier value —which may occur abruptly—, since in the former case it may be the outcome of a slow and progressive degradation, which is more difficult to detect. Regarding degradation, some sensors become less reliable when they reach the end of their useful life.

Incipient faults are the most challenging ones from the validation and imputation point of view, but in practice communication faults and outlier values are the most common ones (instead of incipient faults, they may occur in a daily basis). Thus, from the practitioner point of view this type of faults are the ones that should be addressed first. As this work is developed in the framework of an Industrial PhD to solve a problem of the practitioner, decisions have been generally made based on the premise to solve practical problems, whilst considering the research challenges. Furthermore, although incipient faults are more difficult to detect, they seldomly happen in the actual process and their effects can be minimised by a periodic calibration procedure and following the manufacturer's instructions with regard to sensors replacement frequency.

Finally, another point to note is that a fault could happen in more than one sensor at a time. When the imputation method is based on using data from other sensors in order to predict the missing value for the target one, missing data in those sensors may compromise the performance of the method. Of course, if the number of missing or incorrect data extends to all or almost all available sensors, nothing can be done. The methodology will be first evaluated for a single fault, and then multiple sensor faults will be assessed in the future.

6.3. Validation stage

This section details the proposed methodology for data validation, the first step in the Data Validation and Imputation module in Figure 6.1. The input to this procedure is the case or episode created with raw data gathered from the process sensors. At this first step, if the case $c(t)$ is validated, the imputation step is omitted and the classic CBR cycle is executed to provide the set-points for the situation described by $c(t)$. Conversely, if one or more feature values in $c(t)$ are invalidated, the imputation stage will be performed to provide an estimation for those incorrect or missing values. The result is the new case $\hat{c}(t)$. For the estimated features, the new proposed values are the ones stored in the CB/EB.

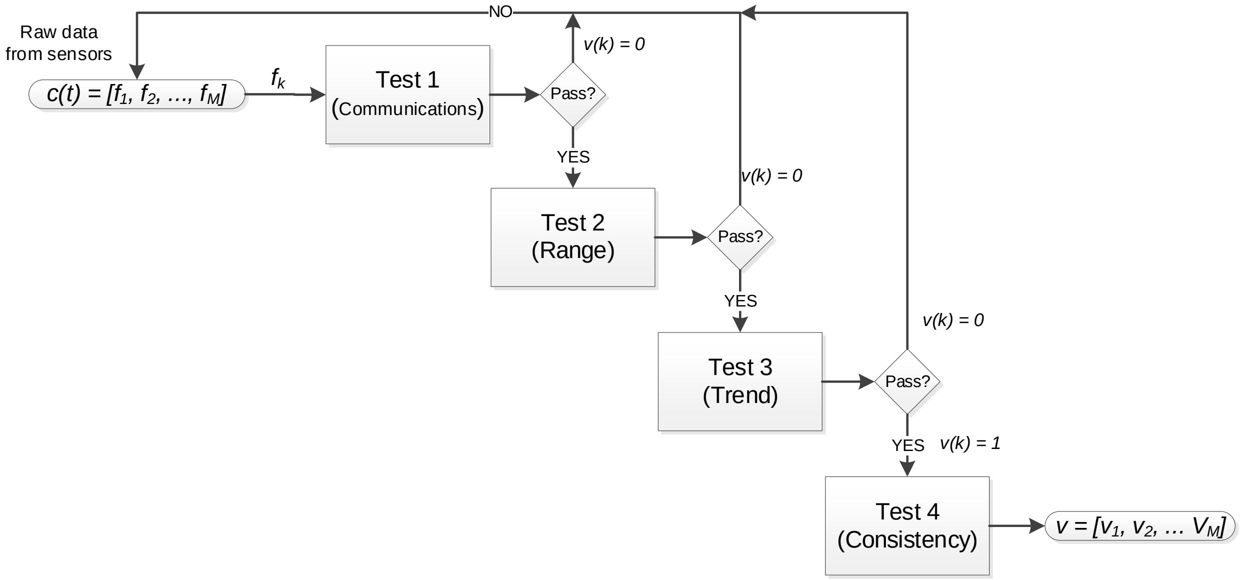


Figure 6.2 Data Validation stage

The data validation process applies a set of tests to the data (Figure 6.2) to determine whether should be validated. Each test is applied to all the features f_k in $c(t)$. If the value for f_k pass all tests, the flag $v(k)$ is set to 1. Otherwise, the flag $v(k)$ is set to 0. Tests 1 to 3 are simple tests that can be applied to any feature. Test 4 is a more complex one which depends on the existence of a spatial relation with other sensor measurements. All tests are described below:

- Test 1 checks whether data is available or not. If the value is missing, it is assumed that the sensor is not working properly or that there is some communication problem from the sensor to the data server, e.g., the OPC server. Communication problems can be detected by comparing two consecutive samples —since in most situations two consecutive samples rarely have the same value—, or by checking the sample time associated to the current sample. If the sample time is not the current one, it means that the value at the current time step have not been obtained. In Figure 6.3 some examples are illustrated. This test is valid for most time-series. When obtaining consecutive equal values is considered a normal process behaviour, this test should be avoided or the time between value changes has to be set to a value greater than one sample.

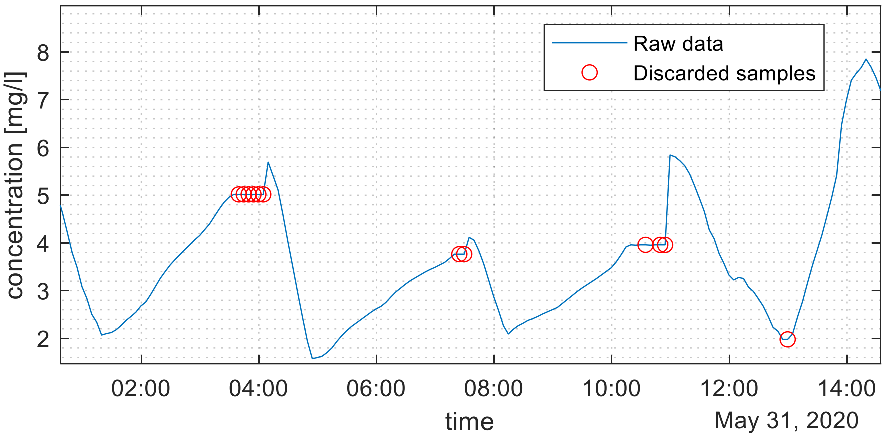


Figure 6.3 Example of dismissed samples in Test 1

- Test 2 verifies whether data are within their physical limits considering the range of the sensors acquiring the measurements. This test can be easily calibrated using the limits from specifications, but also from expert knowledge or historical records. Figure 6.4 shows an example considering the sensors limits.

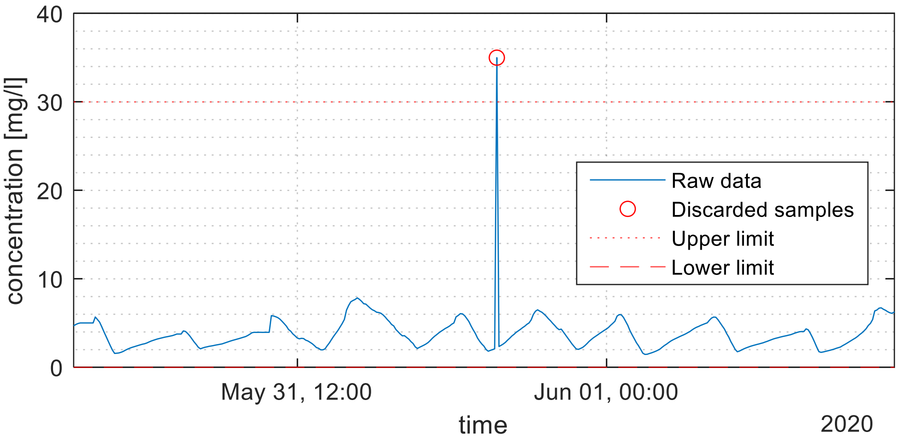


Figure 6.4 Example of a dismissed sample in Test 2

- Test 3 checks whether the change of the data between two consecutive samples is within an expected rate. This test allows detecting unexpected and abrupt changes in the data, e.g., the ammonium concentration in a biological reactor cannot change more than several mg/l per minute (example in Figure 6.5). The accepted range may be calculated from historical validated data or from expert knowledge. Of course, it can be difficult to find a calibration parameter allowing to detect just the incorrect. Thus, depending on the sensor, the configuration can be more or less conservative.

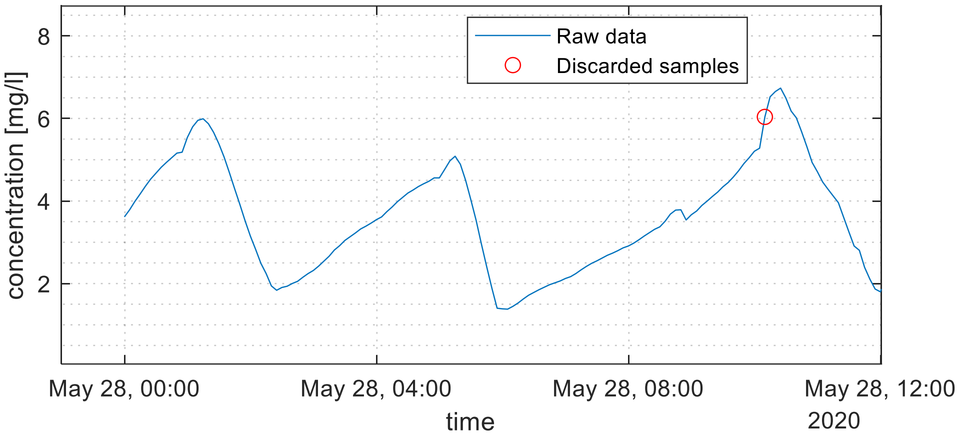


Figure 6.5 Example of a dismissed sample in test 3

- Test 4 checks the consistency —when possible— of the data gathered from a specific sensor using information from other sensors. For example, a flow in a pipe with a valve or a pump. If the valve is closed or the pump is not working, the flow should be zero. Otherwise, the value should be invalidated. This test cannot be applied to all the sensors, but only to those with spatial or causal relations. Other kind of relations, e.g., black box models linking two or more variables, are not considered. Figure 6.6 shows an example for an oxygen sensor using as additional inputs the valve position in the air line and the blower status (on/off). Oxygen is expected to increase when the valve position is open and the blower status is on.

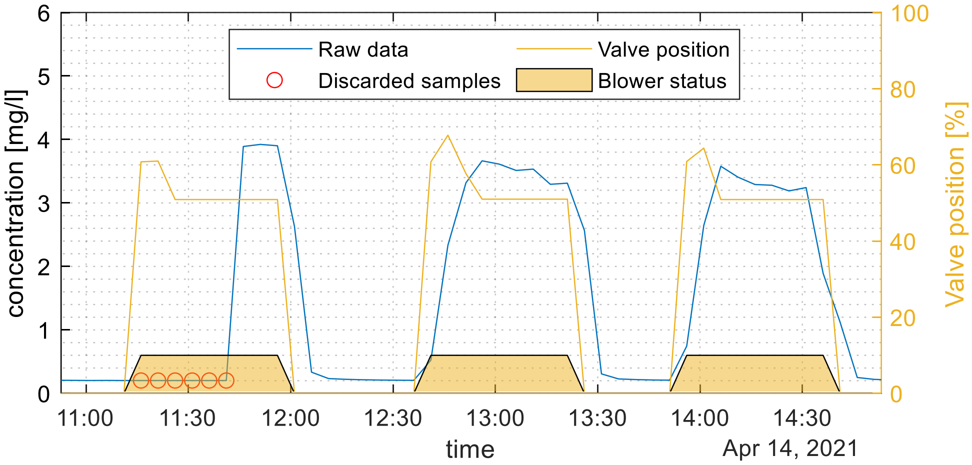


Figure 6.6 Example of a dismissed sample in Test 4

As introduced before, if any of the validation tests in Figure 6.2 is not satisfied, $v(k) = 0$ for the corresponding feature. Then, the imputation stage is executed.

6.4. Imputation stage

The imputation process is activated when incorrect or missing values are detected in the validation stage, i.e., one or more values forming the case $c(t)$ don't pass all tests in the validation stage. The output of the validation stage (v) is used to identify invalid values that should be imputed. As mentioned before, the imputation proposed here is based on using CBR. In Subsection 6.4.1 the imputation based on CBR is described, as well as other classical models such as Auto Regressive (AR) models or Artificial Neural Network (ANN) models. Then, in Subsection 6.4.2 an ensemble of models is proposed with the aim of obtaining a better imputation performance than the one obtained by each individual model separately. Furthermore, the CBR imputation can be compared with some classical prediction models such as AR or ANN. Due to the different nature of the sensors, i.e., different types of time-series, for example, with or without seasonality, the comparison of different models may be interesting, as well as the ensemble of models to enhance the strengths of each individual model.

6.4.1 Imputation models

The proposed imputation model is based on CBR, as shown in Figure 6.1. The CBR principle may be applied to data imputation, using values from similar past situation to replace incorrect or missing data. The imputation method proposed is integrated to a CBR-based IDSS, so the existing CB/EB can be used for imputation purposes. When a value from the current case is incorrect or missing, the available part of the case/episode is used to find the most similar ones in the CB/EB.

As detailed in Section 4.2 from Chapter 4, all the features are numeric, so the Euclidean Distance (ED) similarity measure is also used. To impute the current missing value, the corresponding value obtained from the k most similar cases/episodes can be reused to obtain a new estimated case/episode. The value of k can be equal to one or be greater than one (e.g., the imputed value is calculated as the mean of the set of cases). In addition, both the classical CBR approach detailed in Chapter 4, and the temporal approach described in Chapter 5, are compared. In the first step of the imputation procedure – retrieval (Algorithm 4.1 for cases in Chapter 4 and Algorithm 5.3 for episodes in Chapter 5) –, the distance between the current case/episode with those in the CB/EB can be calculated using expressions ((4-4) or ((5-5), respectively. The difference here is that the current case/episode has some unvalidated values, which will be represented as missing data (NA). For example, one feature is missing in the case in (6-1) and episode in (6-2). In the case, the value of the first feature is missing; hence, if an episode is built from that case, the value of the first feature in the most recent case will be the missing value.

$$c_0 = \left(NA, f_{0,2}, \dots f_{0,N}\right) \tag{6-1}$$

where $f_{0,1}$ to $f_{0,N}$ are the values for features 1 to N in the case c_0.

$$e_0 = (c_{t-J+1}, \dots, c_{t-1}, c_t) = \begin{bmatrix} f_{t-J+1,1} & \cdots & f_{t-J+1,1N} \\ \vdots & \ddots & \vdots \\ NA & \cdots & f_{t,N} \end{bmatrix} \tag{6-2}$$

where c_{t-J+1} to c_t are the cases forming the episode e_0, $f_{t-J+1,1}$ to $f_{t-J+1,1N}$ the values for the features of the oldest case in e_0 and $f_{t,1}$ to $f_{t,N}$ the value for the current case.

To calibrate the retrieval step, the weight w_n of each feature and the episodes length J is calculated to minimize the error between the real value of each feature and the estimated value obtained as the result of the retrieval step, i.e., the corresponding value from the retrieved case/episode. The retrieval process using cases and episodes is illustrated in Figure 5.2 from Chapter 5. In Figure 6.7, based on Figure 5.2, is shown how the imputation is done. Here it is assumed the use of the most similar case/episode, but other approaches can be considered, i.e., the mean of the k most similar cases/episodes.

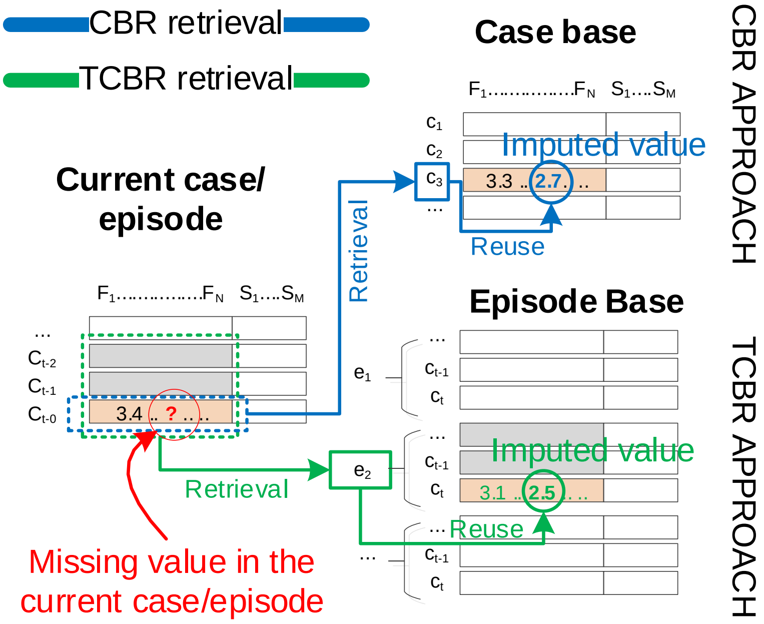


Figure 6.7 Imputation using the retrieved case/episode

The error to minimize, i.e., the cost function F in the optimization problem, is the RMSE, which can be calculated as in equation (6-3):

$$F = \sqrt{\frac{1}{Q} \cdot \sum_{i=1}^{Q} \left(y(i) - \hat{y}(i)\right)^2} \tag{6-3}$$

where Q is the number of samples in the calibration dataset, y is the measured data for a particular feature and $\hat{y}$ is the predicted value for the same feature. Note that the error is calculated for the target feature, i.e., the one to be estimated in case of missing value, and only for the most recent case in the episode, not for the most recent case and the preceding ones. Thus, episode length L does not appear in the cost function F. Considering a single fault, a set of w_n weights is tuned for each feature. For example, for M features, a set of M weights' vectors will be necessary. To calculate these weights a method for solving constrained optimization problems can be used. The optimization problem can be defined as in (6-4)

$$\min_{w} F(w)$$

Subject to:

$$\sum_{n=1}^{N} w_n = 1 \qquad (6\text{-}4)$$

Where:

$$0 \leq w \leq 1$$
$$1 < L < L_{max}$$

where w is the weights vector and $F(w)$ is the cost function to be minimized (here the RMSE expressed in (6-3). The weight for a particular feature must be a value between 0 and 1, and the sum of the weights vector must be 1. Episodes length must be a value between 1, i.e., single case, and L_{max}, which is the maximum length considered.

This problem is solved using the Global Optimization Toolbox in MATLAB, which provides a set of functions to search for global solutions to problems than contain multiple maxima or minima. In particular, the Genetic Algorithm (GA) function is used. The multiple faults problem will be addressed in the future. In the framework of this thesis the focus is on single faults.

The proposed CBR-based imputation method is compared with other classical methods used for time-series modelling, such as AR models or ANN models. *AR models* (Brockwell and Davis, 1991) specify that the output variable depends linearly (in the case of linear AR models) on a combination of previous values, as specified in equation (6-5):

$$y(t) = \sum_{i=1}^{n} a_i \, y(t-i) \qquad (6\text{-}5)$$

where y is the target time series (the one to be predicted), n is the number of considered past values for y time series and a_i are the model coefficients.

The AR model can also consider other input variables or exogenous variables that can be related with the target one (ARX model). This model can be represented by equation (6-6):

$$y(t) = \sum_{i=1}^{n} a_i \, y(t-i) + \sum_{j=1}^{m} b_j \, u(t-j) \tag{6-6}$$

where y is the target time series (the one to be predicted), u are the external exogenous variables, n and m are the number of considered past values for y and u time series and a_i and b_i are the model coefficients. The MATLAB tool Regression Learner is used to find AR models.

*Artificial Neural Network (ANN) models (*Walczak and Cerpa, 2003) are inspired by biological nervous systems. As in the nature, the connections between different nodes or artificial neurons determine the network behaviour. Here, an ANN is trained to solve a non-linear time-series problem in order to predict future values. Like in the AR models, present and past values from other time-series different than the target one can be considered (6-7).

$$y(t) = f\big(y(t-1), \dots, y(t-n), u(t-1), u(t-n\,)\big) \tag{6-7}$$

where y is the target time series (the one to be predicted), u are the external exogenous variables and n is the number of considered past values for y and u time series.

The MATLAB tools Neural Net Time Series is used to train Neural Networks to solve the problem defined in (4.7). Figure 6.8 shows the structure of one of these Neural Networks.

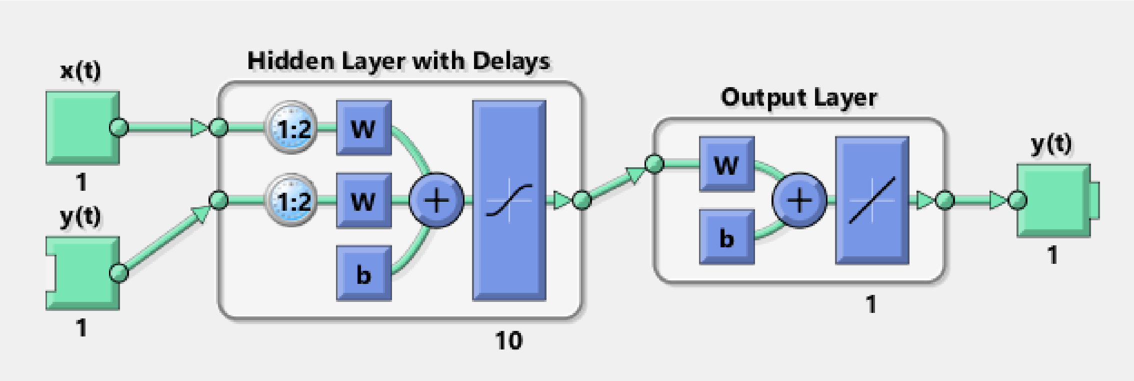


Figure 6.8 Non-linear autoregressive with external (exogenous) input (NARX) Neural Network

6.4.2 Ensemble of models

Taking advantage of the calibration of multiple models to compare its performance, it is proposed the use of ensemble methods to combine them. An ensemble method integrates the combination of the predictions of different models to obtain a single prediction (Dietterich 2000). This combination aims to achieve a better prediction than the one provided by individual models, hence improving the performance.

Here, four methods have been considered in order to implement an ensemble model for data imputation: voting (V), weighted voting (WV), Linear Regression Meta-Predictor (LR-MP) and Artificial Neural Network Meta-Predictor (ANN-MP).

In the voting method (Figure 6.9), different models are trained with the same training dataset. In this method, the model with the best accuracy in the prediction, is selected. The RMSE of each model is evaluated in a moving horizon window. At time step t_k, the RMSE for the model j is calculated as in (6-8):

$$\mathrm{RMSE}_j(k) = \sqrt{\frac{1}{m}\sum_{i=k-m}^{k-1} e(i)^2} \tag{6-8}$$

where m is the length of the evaluation window and e(i) is the error between the prediction and the measured value at time step i.

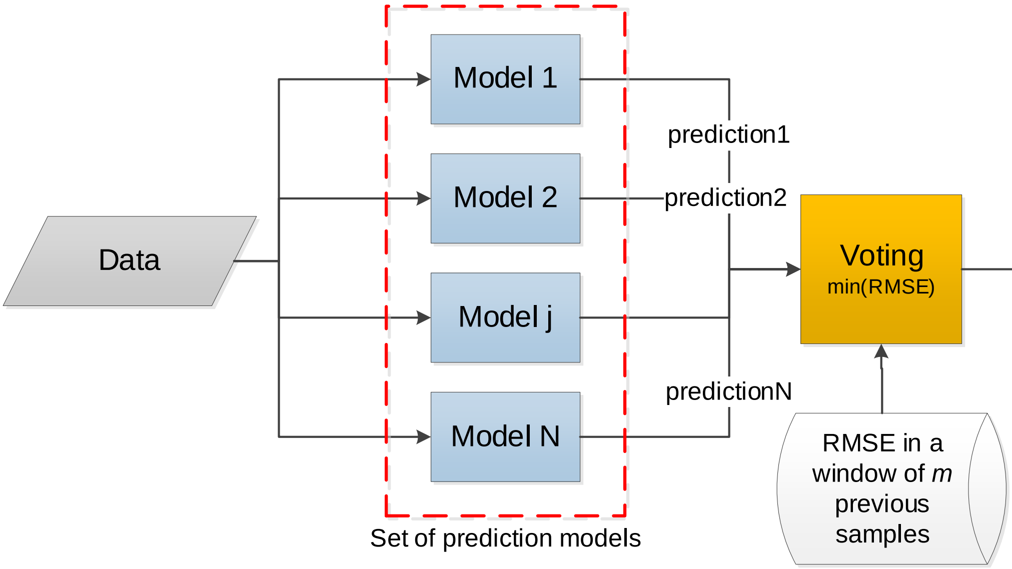


Figure 6.9 Voting algorithm

The weighted voting method (Figure 6.10) is similar to the voting method, but in this case, the selected model is not the one with the lowest RMSE in the evaluation window. Here, each model prediction is weighted by its RMSE value, i.e., the lower error predictions will have more weight in the final prediction (6-9):

$$\hat{y}(k) = \sum_{j=1}^{N}\left(w_j \cdot \hat{y}(k)_j\right) \tag{6-9}$$

where N is the number of imputation models, $\hat{y}(k)_j$ is the prediction for the model j and w_j is the weight given to the model j to obtain the final prediction $\hat{y}(k)$. The model with the smaller RMSE will have the highest weight. The weight w_j can be calculated as in (6-10):

$$w_j = \frac{1}{RMSE_j(k) \cdot \sum_{j=1}^{N} RMSE_j(k)} \tag{6-10}$$

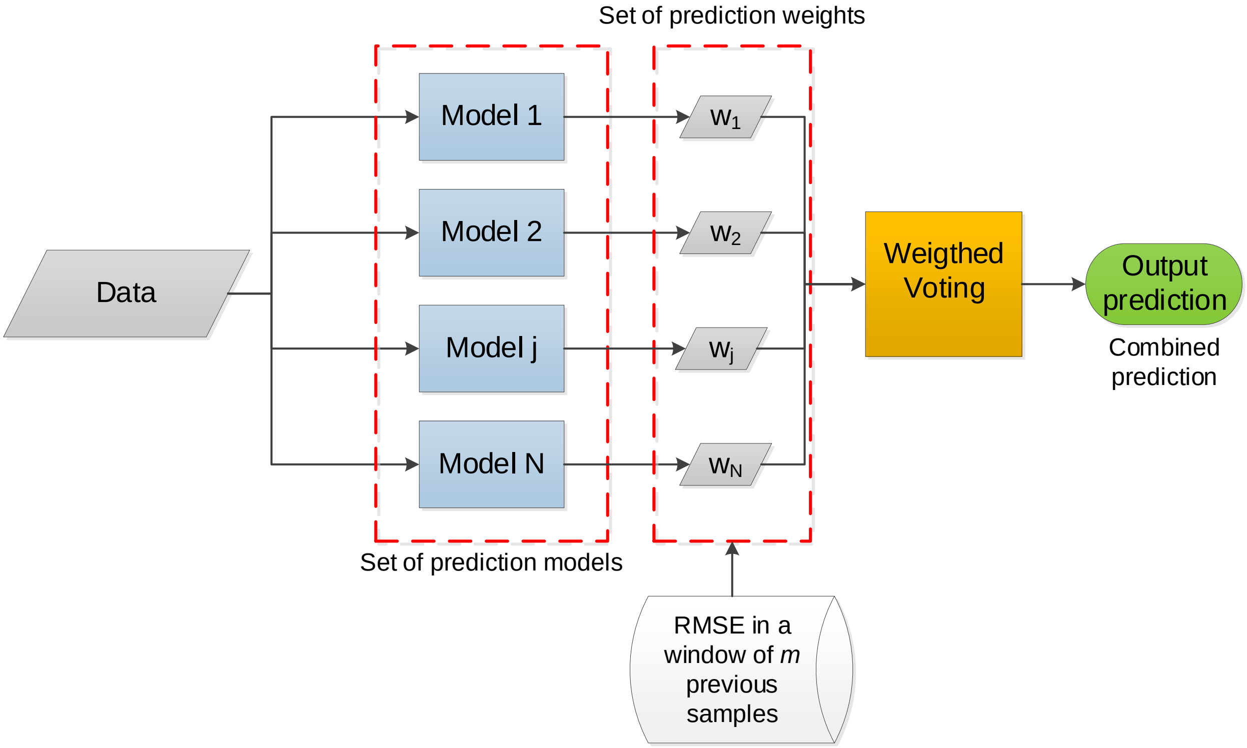


Figure 6.10 Weighted voting algorithm

In the case of LR-MP and the ANN-MP ensemble methods, a stacking algorithm (Figure 6.11) is used to provide the prediction. The stacking considers not only the measured data, but also the predictions of each individual model to train a meta-predictor in order to provide a prediction based on the different predictions generated by each model. The meta-predictor attempts to learn how to best combine the input predictions to find a better prediction output. As shown in Figure 6.11, the MP training dataset consist of the Training dataset, including all input parameters I_K, as well as the desired output O_1, extended with the prediction obtained by each model from 1 to N. The MP model is trained using this extended dataset to obtain the prediction P_{O1}. In the same way, when new data is available, the prediction using each individual model is obtained and then, the new data set is extended with those prediction as the input of the MP model to generate the final prediction.

The LR-MP is obtained training a linear regression model, while the ANN-MP is trained using a dynamic neural network. For the implementation of all models, MATLAB software has been used, in particular System Identification toolbox, Statistics and Machine Learning toolbox and Deep Learning toolbox.

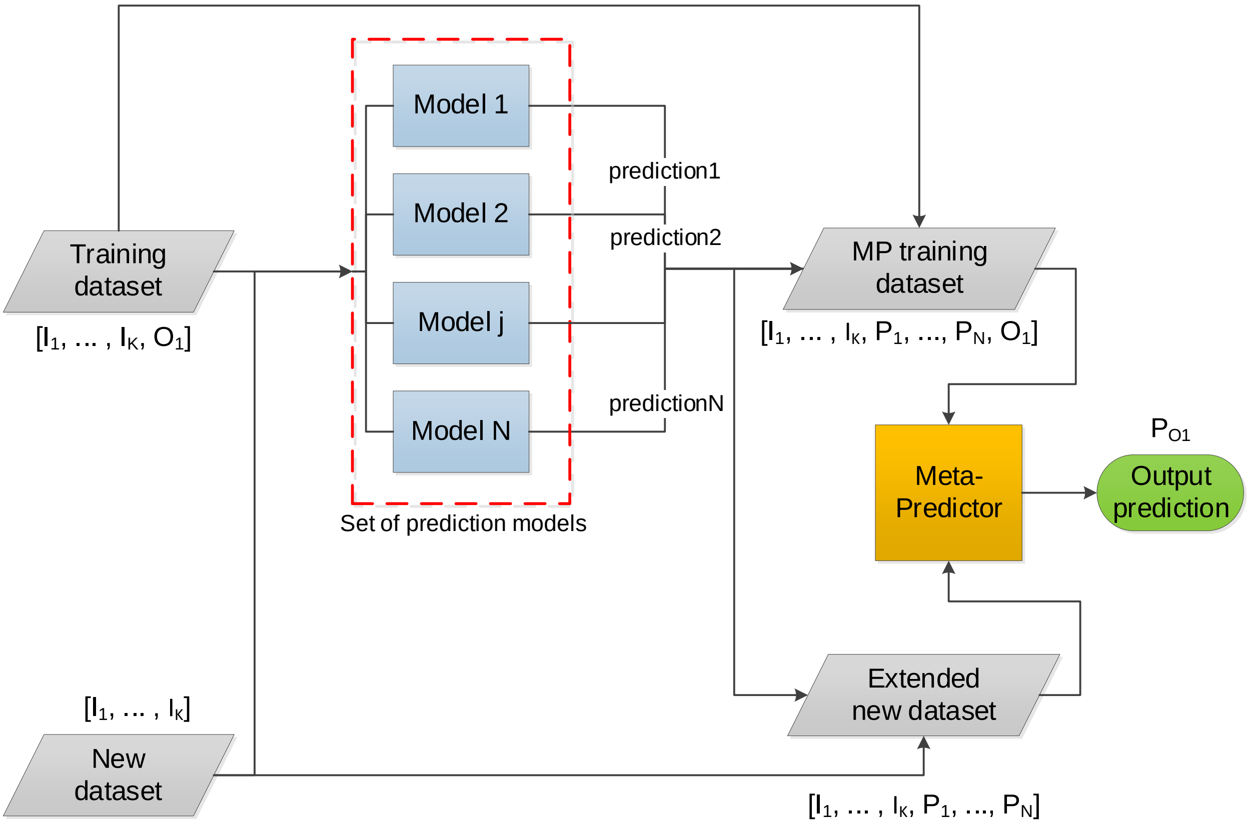


Figure 6.11 Stacking algorithm

6.5. Conclusions

In **Chapter 6**, a data validation and imputation method is proposed. Since the performance of data-driven proposals in Chapters 5 and 6 is highly dependent on data quality, a Data Validation and Imputation module is proposed to be integrated in the developed tool. This module aims to overcome the problem with short to medium length faults, which are the most frequently occurring in real situations –isolated one-sample faults to six-sample faults (from five to 30 minutes).

For the data validation process a set of validation tests is performed to time-series, following a sequential scheme. The first three test are basic tests to check, first, the availability of a value at the current time step, and then, whether this value is within a coherent range and trend. Finally, the consistency of the value can be checked using relations between sensors. Regarding the imputation process, the calibration of a TCBR-based method using GA is proposed. In addition, different ensemble strategies to combine the TCBR-based solution with other classic black-box methods such as ANN or LR are proposed to improve the overall performance.

7. Experimental Results

This chapter presents the results in line with the different objectives of this work, detailed in Section 1.2. In Section 7.1 the case study is described. In Section 7.2, the implementation of the IDSS based on the concept proposed in Chapter 3 is presented. Section 7.3 shows how the implemented tool have been deployed in two real facilities and the results of its operation during several months. Finally, Section 7.4 presents the results of the last contribution of this work, the validation and imputation module.

7.1. Case study

7.1.1 Consorci Besòs Tordera

Consorci Besòs Tordera (CBT) is a local water administration composed of 64 municipalities in five different regions of Catalonia (Barcelonès, Moianès, Osona, Vallès Occidental and Vallès Oriental) with a population of about 2.400.000 inhabitants (Figure 7.1 and Figure 7.2). Other members are the Diputació de Barcelona (DIBA), the Àrea Metropolitana de Barcelona (AMB), the County Council of Vallès Oriental and the Consortium for Waste Management of Vallès Oriental.

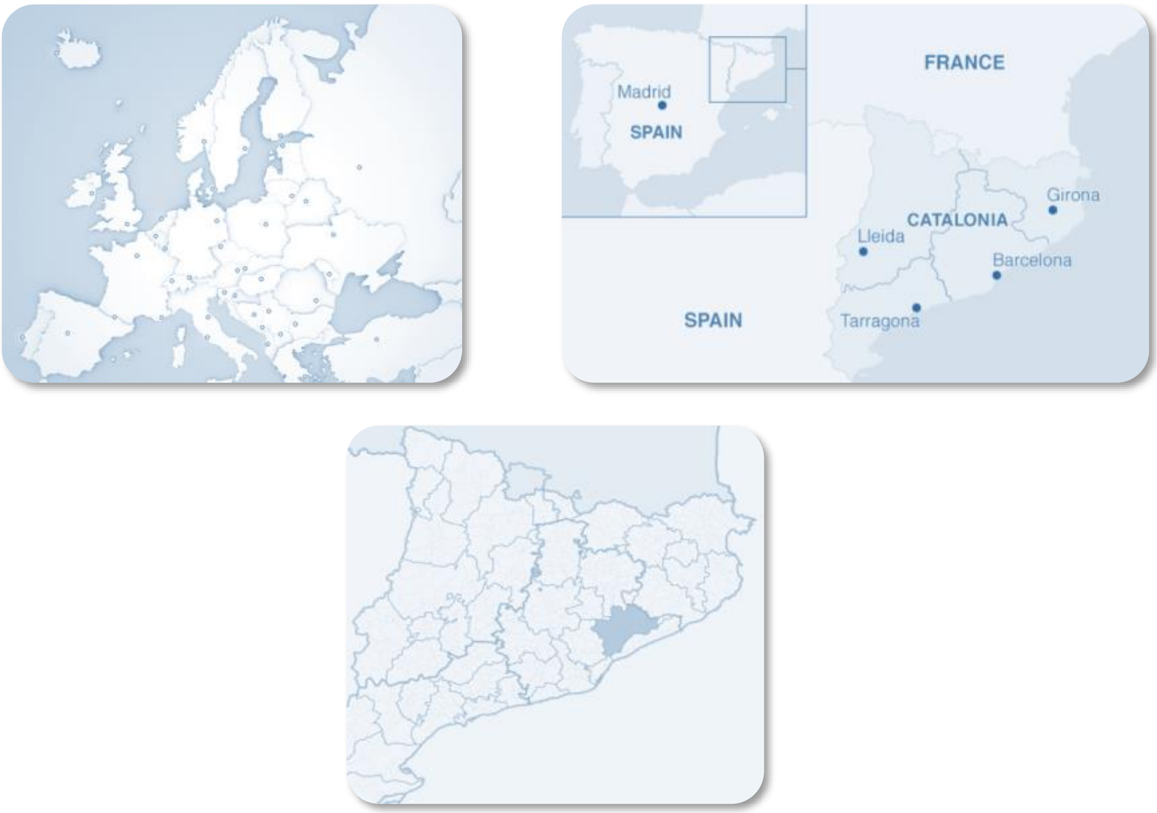


Figure 7.1 CBT location and area of action

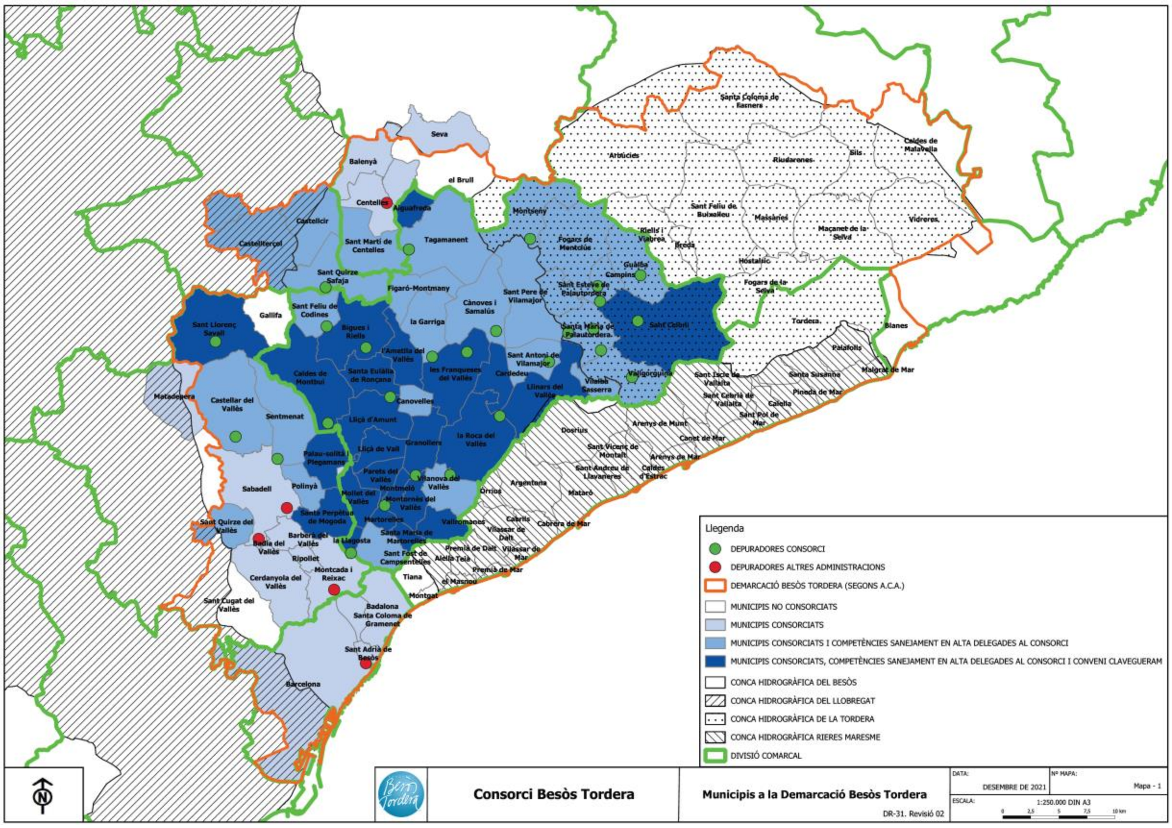


Figure 7.2 CBT managed area, including municipalities limits and WWTPs

The main objective of CBT is preserving and improving the good health of the rivers in its area of actuation. To get this objective, the lines of action of CBT are:

- The sanitation, from municipal sewers to main sewers, pumping stations and waste water treatment plants: cleaning and conservation services of 1300 km of municipal sewers, 300 km of main sewers, 27 WWTPs and 50 pumping stations or the control of industrial discharges, with almost 6000 potentially polluting establishments.
- Improvement and conservation of the river basin: projects for the recovery, restoration, conservation and cleaning of the river basin and river paths.
- Research and application of the best available technologies, as well as the newest and most incipient ones for the continuous improvement of the sanitation systems and to best meet the requirements for the current and future challenges.
- Environmental education: activities addressed to all segments of population and with different orientations: informative, educational or technical. It can be highlighted some examples, like the educational program “Discover the River”, the awareness campaign “The River in your hands” or technical conferences like the conference “Prevention and control of odours in sanitation”.
- The support to all consortium members: Support in different activities like drafting of master plans, restoration of river environment, in the execution of sewage system works or removing wastewater discharges in the environment.

7.1.2 Santa Maria de Palautordera WWTP

Santa Maria de Palautordera WWTP is located in the area of the Tordera river (Figure 7.3). Its design capacity is 3225 m^3/day. The pollution load comes mainly from urban wastewater and it is of about 18000 population equivalent (PE). The water line is composed of a primary treatment, two biological reactors and two secondary clarifiers. The sludge line includes thickening and dewatering processes. Currently, the plant is operated with the primary treatment, two biological reactors and two secondary clarifiers, and it is treating an input flow of about 85 m^3/hour with the characterization described in Table 7.1, obtained from operational data of period 2019.

Table 7.1 Influent characterization for Santa Maria de Palautordera WWTP

Parameter	**Units**	**Concentration**	**Range**
Suspended Solids (SS)	mg/l	136	[12; 1090]
Chemical Oxygen Demand (COD)	mg/l	528	[138; 2610]
Biological Oxygen Demand (BOD)	mg/l	294	[65; 1350]
Nitrogen (N)	mg/l	58	[18; 127]
Ammonium (NH_4)	mg/l	42	[13; 80]
Nitrate (NO_3)	mg/l	0.7	[1; 4]
Phosphorus (P)	mg/l	6.3	[2; 22]
Conductivity	µS/cm	829	[485; 1119]
pH	-	8	[7; 9]

The aeration system consists of three blowers: the main blower and two additional backups. With the current configuration, the main blower can be combined with one backup blower to reach the desired oxygen concentration in the biological reactor. Backup blowers' operation is combined in order to balance operating hours of each element. Each blower has an integrated frequency converter, so the air flow introduced in the biological reactor can be controlled using this element. The air is introduced in the reactor in two different opposite points. Each air inlet is equipped with a solenoid-controlled valve to control the air flow. Thus, the set-points that can be used to control the biological process are the *dissolved oxygen (DO)* set-point, the *air pressure* set-point and the *nitrification / denitrification* command. These set-points are used by a lower-level controller, i.e., a Proportional Integral Derivative (PID) controller, for blowers' speed regulation and air valves position. Figure 7.4 and Table 7.2 show all the available measured variables in the plant. The sample time for all online measurements is set to 5 minutes, which considering the dynamics of the processes has been considered appropriate for the supervision and control of the system. In the case of *Santa Maria de Palautordera* WWTP SCADA system, this value is unknown, since implements old technology and the data can be only displayed graphically, but not retrieved.

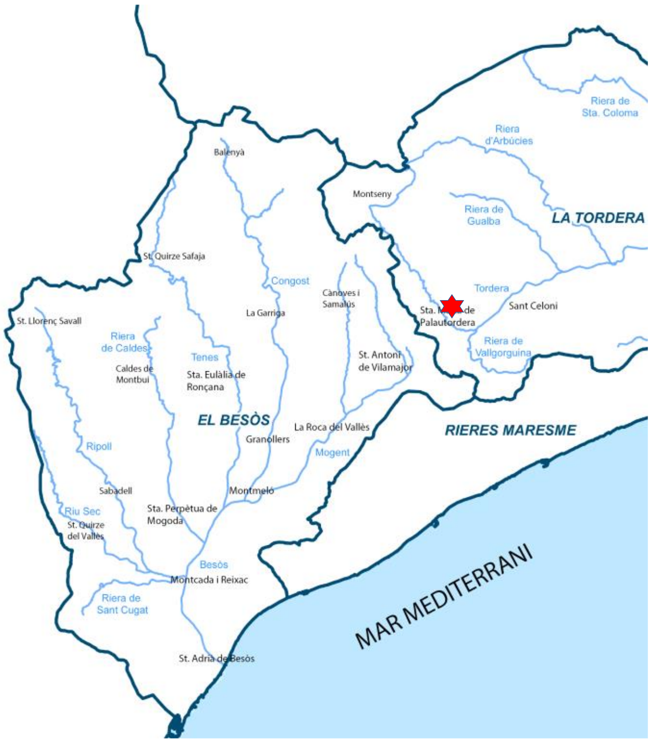


Figure 7.3 Location of Santa Maria de Palautordera WWTP in the CBT river basin

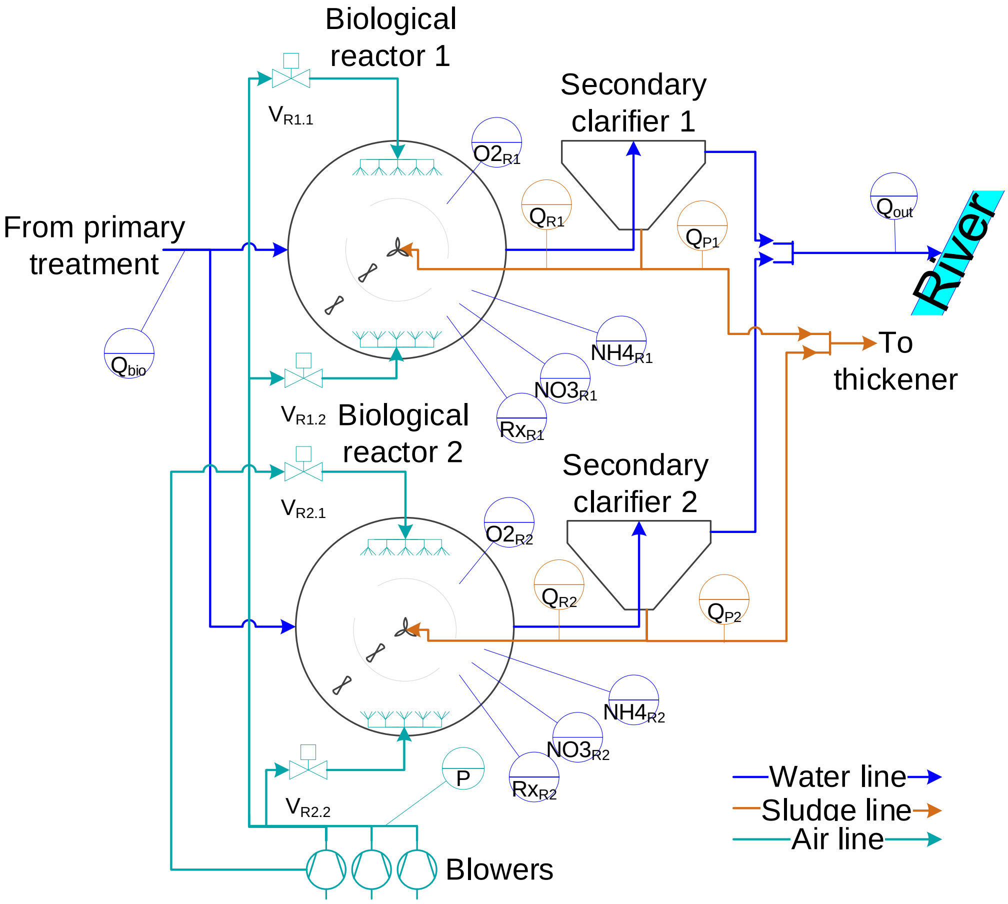


Figure 7.4 Santa Maria de Palautordera WWTP layout

Table 7.2 Available sensors for Santa Maria de Palautordera WWTP

Sensor	Units	Range	Sensor Id.
Plant input flow	m^3/h	[0; 2000]	Q_{bio}
Plant output flow	m^3/h	[0; 2000]	Q_{out}
Sludge recirculation flow	m^3/h	[0; 100]	Q_{R1}, Q_{R2}
Purge flow	m^3/h	[0; 100]	Q_{P1}, Q_{P2}
Ammonium concentration	mg/l	[0; 30]	$NH4_{R1}$, $NH4_{R2}$
Nitrate concentration	mg/l	[0; 30]	$NO3_{R1}$, $NO3_{R2}$
Dissolved oxygen	mg/l	[0; 6]	$O2_{R1}$, $O2_{R2}$
Redox	mV	[-500; 500]	Rx_{R1}, Rx_{R2}
Air pressure sensor	mbar	[0; 600]	P
Air valves position	%	[0; 100]	$V_{R1.1}$, $V_{R1.2}$, $V_{R2.1}$, $V_{R2.2}$

The former control system operating in this WWTP (i.e., before the integration of the IDSS presented here) was based on the redox measurement to regulate nitrification and denitrification phases, combined with open-loop fixed timers for nitrification and denitrification, set by the operator experience. To improve the control of nitrification and denitrifications phases ammonium and nitrate sensors were installed in each biological reactor, as well as new air valves to better regulate the air flow, because the existing ones were not suitable for a precise flow regulation. In a first stage the plant was working with only one biological reactor and with two secondary clarifiers. Then, due to an increase of the aeration needs to better reduce the polluting load the second biological reactor was activated. Considering all these changes, available historical data is not useful to initialize the reasoning system, but the expert's knowledge is coded in a set of rules to initialize the RBR module. This set of rules should cover the generation of all required set-points for the control of the process mentioned above. In particular, these rules include: first, the control of the nitrification and denitrification phases; second, the increase or decrease of the DO setpoint and; finally, the increase or decrease of the pressure setpoint. Some of these rules are shown in (7-1) as an example:

```
Rule 1: IF NH4 >= NH4_max OR
            (IF NH4 > NH4_min AND N == 1) THEN Nitrification
Rule 2: IF NH4 <= NH4_min OR
            (IF NO3 > NO3_max AND NH4_24h < 4) OR                    (7-1)
            (IF NH4 > NH4_min AND NH4 <= NH4_max AND N == 0)
            THEN Denitrification
Rule 3: IF ΔNH4 > -ΔNH4_min THEN Increase O2
```

7.1.3 Castellar WWTP

Castellar WWTP is located in the area of the Ripoll river (Figure 7.5). Its design capacity is 8000 m^3/day. The pollution load comes mainly from urban wastewater and it is of about 76500 population equivalent (PE). The water line is composed of a primary treatment with two decanters, two biological reactors and three secondary clarifiers. The sludge line includes thickening and dewatering processes. The process scheme is depicted in Figure 7.6. Currently, the plant is operated with the one primary decanter, two biological reactors and three secondary clarifiers, and it is treating an input flow of about 330 m^3/hour with the characterization described in Table 7.3, obtained from operational data of period 2019. The second primary decanter is used to laminate the inflow.

Table 7.3 Influent characterization for Castellar del Vallès WWTP

Parameter	Units	Concentration	Range
Suspended Solids (SS)	mg/l	299	[129; 847]
Chemical Oxygen Demand (COD)	mg/l	636	[265; 1050]
Biological Oxygen Demand (BOD)	mg/l	339	[170; 640]
Nitrogen (N)	mg/l	66	[37; 107]
Ammonium (NH_4)	mg/l	47	[26; 80]
Nitrate (NO_3)	mg/l	1	[0; 3]
Phosphorus (P)	mg/l	10	[4; 69]
Conductivity	µS/cm	nd	nd
pH	-	8	[6; 9]

The aeration system consists of five blowers of different types: two blowers with variable speed, two with two fixed speeds (slow and fast) and one with one speed. With the current configuration, different combinations of two blowers can be used to reach the desired oxygen concentration in the biological reactor avoiding overpressure problems. Blowers are combined in order to balance operating hours of each element. Each biological reactor is composed of three chambers where the air is introduced. Each chamber is equipped with a solenoid-controlled valve and an oxygen sensor. In the last chamber of each biological reactor there are an ammonium sensor and a nitrate sensor. Thus, the set-points that can be used to control the biological process are the *DO* set-point, the *air pressure* set-point and the *nitrification / denitrification* command. These set-points are used by a lower-level controller, i.e., a Proportional Integral Derivative (PID) controller for blowers' speed regulation and air valves position. The current available sensors are listed in Table 7.4.The sample time for all online measurements is set to 5 minutes, which has been considered appropriate for the supervision and control of the system. Usually, SCADA systems register data with a higher frequency. In the case of *Castellar del Vallès* WWTP the SCADA system sample time depends on the sensor, usually between 1second and 5 minutes.

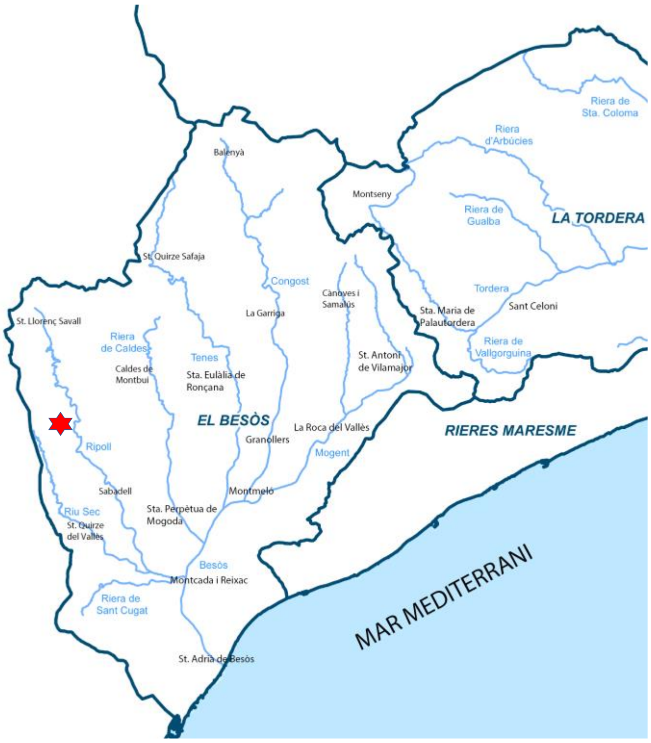


Figure 7.5 Location of Castellar WWTP in the CBT river basin

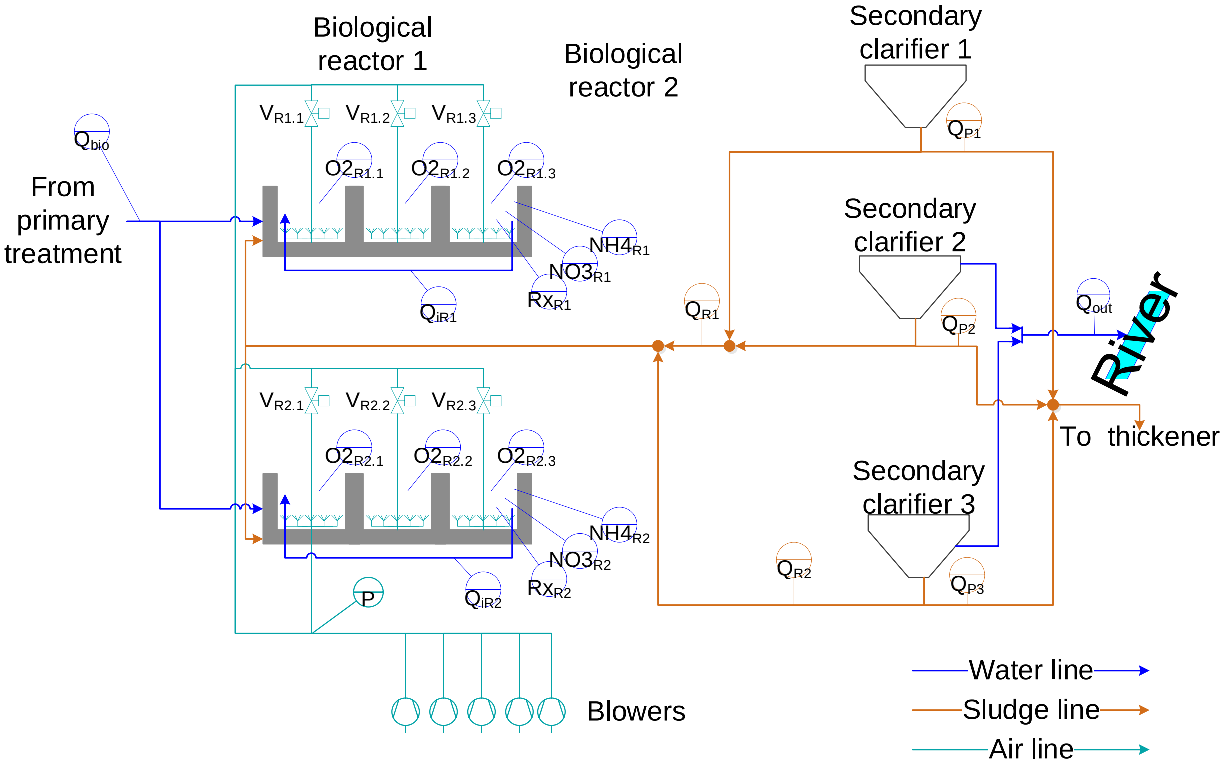


Figure 7.6 Castellar WWTP layout

The former control system operating in this WWTP (i.e., before the integration of the IDSS presented here) was based in a unique ammonium measurement in the outlet of both biological reactors to regulate nitrification and denitrification phases, combined with open-loop fixed timers for nitrification and denitrification, set by the operator experience. Air valves was manual valves and there is only one oxygen sensor in each biological reactor. Thus, valves were positioned manually by plant operators and airflow couldn't be controlled as accurately as with an automated system. For this reason, available historical data is not useful to initialize the reasoning system, but the expert's knowledge is coded in a set of rules to initialize the RBR module. This set of rules should cover the generation of all required set-points for the control of the process mentioned above. In particular, these rules include, first, the control of the nitrification and denitrification phases; second, the increase or decrease of the DO setpoint and finally, the increase or decrease of the pressure setpoint. Rules for this system are similar to the ones in (7-1), only changing the values of some configurable parameters

Table 7.4 Available sensors of Castellar del Vallès WWTP

Sensor	**Units**	**Range**	**Sensor Id.**
Plant input flow	m3/h	[0; 2000]	Q_{bio}
Plant output flow	m3/h	[0; 2000]	Q_{out}
Sludge recirculation flow	m3/h	[0; 100]	Q_{R1}, Q_{R2}
Internal recirculation flow	m3/h	[0; 100]	Q_{iR1}, Q_{iR2}
Purge flow	m3/h	[0; 100]	Q_{P1}, Q_{P2}, Q_{P3}
Ammonium concentration	mg/l	[0; 30]	$NH4_{R1}$, $NH4_{R2}$
Nitrate concentration	mg/l	[0; 30]	$NO3_{R1}$, $NO3_{R2}$
Dissolved oxygen	mg/l	[0; 6]	$O2_{R1.1}$, $O2_{R1.2}$, $O2_{R1.3}$, $O2_{R2.1}$, $O2_{R2.2}$, $O2_{R2.3}$
Redox	mV	[-500; 500]	$Rx_{R1,}$ Rx_{R2}
Air pressure sensor	mbar	[0; 600]	P
Air valves position	%	[0; 100]	$V_{R1.1}$, $V_{R1.2}$, $V_{R1.3}$, $V_{R2.1}$, $V_{R2.2}$, $V_{R2.3}$

7.1.4 Data collection

The data collection architecture is depicted in Figure 7.7. Sensors are physically connected to the plant Programmable Logic Controller (PLC), which is a modular hardware device for the control of processes. The method depends on the technology available for each sensor. Then, measured values are transferred to be visualized and stored in the Supervisory Control and Data Acquisition (SCADA) system using the communications standard Ole for Process Control (OPC), which is based on a client-server architecture very extended in the industry. The same structure is valid to provide data to the IDSS. In Figure 7.8 relevant information provided by the OPC client is highlighted: the variable ID, which is the ID used in the PLC program, the last value obtained, the time stamp of this value and the quality of the value. The quality value is used to represent the validity of the value, or in other words, whether or not the OPC client can trust the value. The quality is divided in three categories: good, bad or uncertain. For each category, additional information describing the quality in more detail can be obtained.

Then, acquired data by the SCADA software is stored its local database. One of the weaknesses of this SCADA system is that the database where data is stored is a proprietary database that cannot be accessed from another application to retrieve data. Data can be easily visualised inside the SCADA system, but it is usually time limited, e.g. data from the last 2 years, and difficult to be exported for other uses. For each measure, one monthly file is generated for the whole 2 years period, i.e., 24 files for each sensor or variable derived from them. These files can be, first, converted to CSV format, and then aggregated in a unique CSV file using a dedicated applet for this purpose. Each CSV file contains a header with information about all original files (sample period, file location, variable units, start and end time, etc), and then, a two-column comma-separated list with the timestamp and the corresponding measured value.

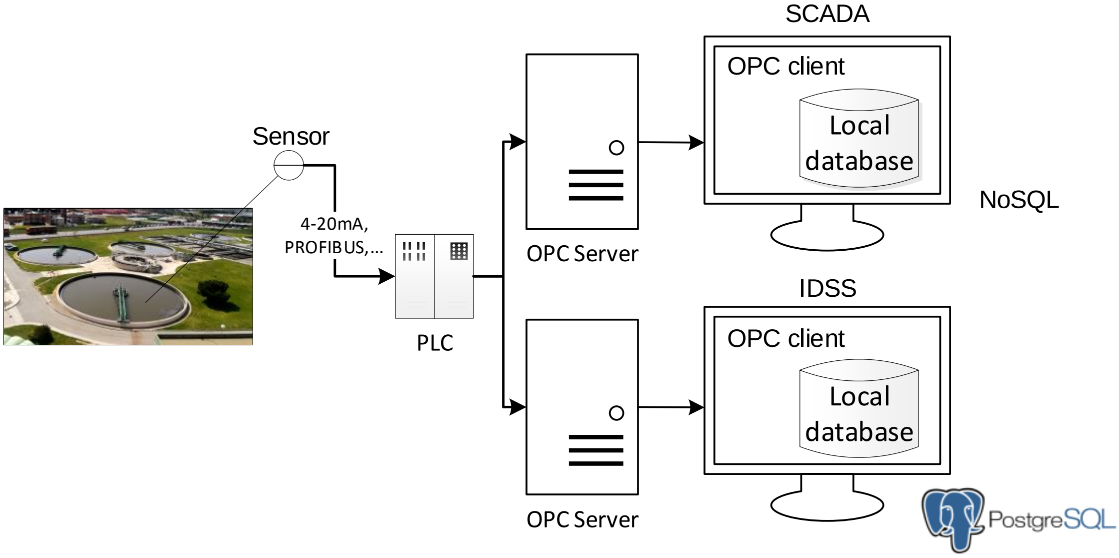
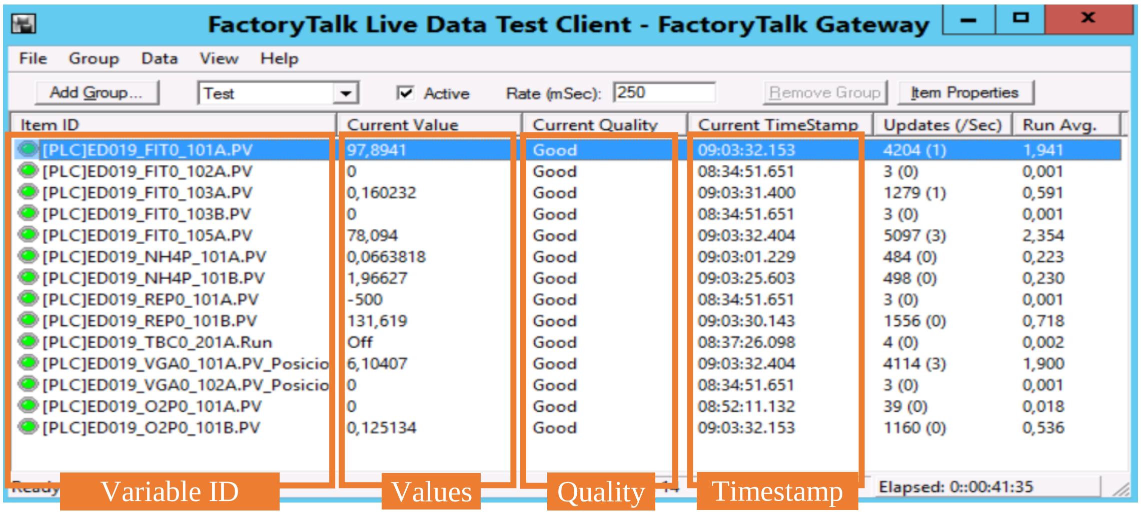


Figure 7.7 Data collection architecture



Test

☑ Active

Rate (mSec): 250

Item ID	Current Value	Current Quality	Current TimeStamp	Updates (/Sec)	Run Avg.
[PLC]ED019_FIT0_101A.PV	97,8941	Good	09:03:32.153	4204 (1)	1,941
[PLC]ED019_FIT0_102A.PV	0	Good	08:34:51.651	3 (0)	0,001
[PLC]ED019_FIT0_103A.PV	0,160232	Good	09:03:31.400	1279 (1)	0,591
[PLC]ED019_FIT0_103B.PV	0	Good	08:34:51.651	3 (0)	0,001
[PLC]ED019_FIT0_105A.PV	78,094	Good	09:03:32.404	5097 (3)	2,354
[PLC]ED019_NH4P_101A.PV	0,0663818	Good	09:03:01.229	484 (0)	0,223
[PLC]ED019_NH4P_101B.PV	1,96627	Good	09:03:25.603	498 (0)	0,230
[PLC]ED019_REP0_101A.PV	-500	Good	08:34:51.651	3 (0)	0,001
[PLC]ED019_REP0_101B.PV	131,619	Good	09:03:30.143	1556 (0)	0,718
[PLC]ED019_TBC0_201A.Run	Off	Good	08:37:26.098	4 (0)	0,002
[PLC]ED019_VGA0_101A.PV_Posicio	6,10407	Good	09:03:32.404	4114 (3)	1,900
[PLC]ED019_VGA0_102A.PV_Posicio	0	Good	08:34:51.651	3 (0)	0,001
[PLC]ED019_O2P0_101A.PV	0	Good	08:52:11.132	39 (0)	0,018
[PLC]ED019_O2P0_101B.PV	0,125134	Good	09:03:32.153	1160 (0)	0,536

Elapsed: 0::00:41:35

Figure 7.8 OPC client window highlighting relevant items read from the PLC

Since the approach presented in this thesis is data-based, historical data availability is crucial, as well as to enable the access to data to other applications or methods. For this reason, and to avoid data accessibility problems in the future, it has been a design requirement to use an open-source database in the deployed IDSS so as not to be dependent of the SCADA system.

In Section 6.2 the data quality problem has been described from a general point of view. As stated in the previous chapter, the data validation and imputation approach is focused to solve the most common type of faults, namely communication problems and outlier values, which usually last a few samples.

Figure 7.9 shows raw data from a sensor during a period of three months. Considering the validation tests described in Section 6.3, some samples have been invalidated. The number of invalidated samples is about 2%. Most of the time (95%), the length of the invalid sample periods is from one to six consecutive samples –i.e. 30 minutes considering a sample time of 5 minutes–. The 85% of invalid values are isolated samples.

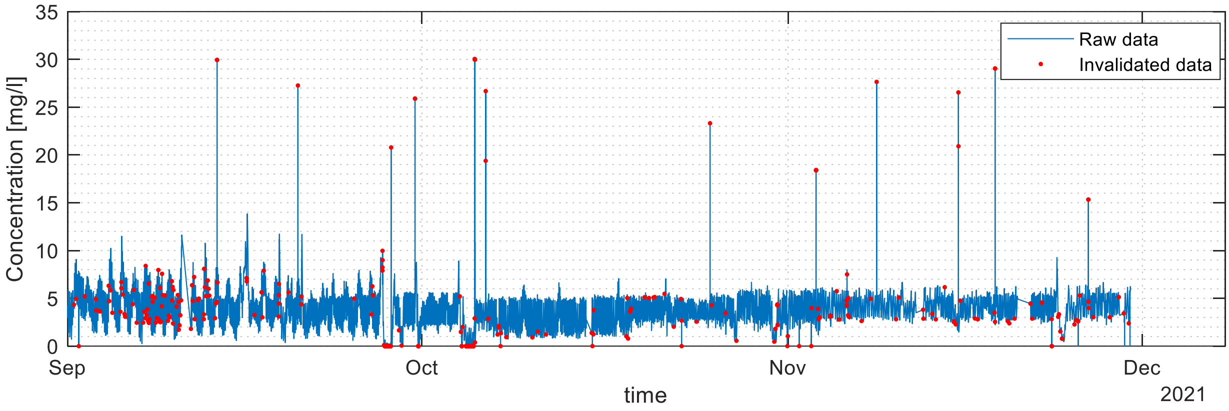


Figure 7.9 Real faults during a period of three months

Accordingly, the methodology proposed in Sections 6.2 and 6.3 is focused on short and medium length faults, i.e., from one to six samples. Longer faulty periods –from hours to days–, are usually due to sensor failures. This type of faults will be not considered here. From the operation point of view, no action was taken for short or medium faults before the implementation of this methodology, which lead to unreliable decision making. Regarding long faults, plant operators have available alternatives to control the process without the damaged sensor, although they are usually less efficient than using the automatic control with all sensors available.

7.2. IDSS Implementation

The first result of this thesis is the implementation of an IDSS and its operation in two real facilities, as described in the results Section 7.3. The implementation has been inspired in the concept presented in Chapter 3 and the methodology proposed in Chapters 4 and 5. Section 9.2 proposes some future work lines regarding this implementation.

7.2.1 Development environments research

In order to develop the framework described in Chapter 3, the use of visual workflows is proposed. To this end, the use of graphical programming environments provides some advantages in relation to traditional languages like C or Java (Johnston et al., 2004), e.g., reusability and understandability of the code, modularity and flexibility, intrinsic parallelism, easy debugging or faster prototyping and development. To choose a valid developing environment is necessary to define the desired specifications. The use of graphical programming languages seems appropriate to create standard and reusable toolboxes. The idea of using open-source languages and tools is also explored. Methods and algorithms needed are all related to data mining: Table 7.5 shows a non-exhaustive list of useful data science techniques and algorithms that can be useful for the development proposed in this work. In (Gibert et al., 2010, 2018a) an overview of different data mining techniques and choosing criteria are presented. (Gibert et al., 2018b) provides an insight to the data science field and its challenges in the environmental domain.

Table 7.5 Data mining techniques and reasoning methods

Data Filtering/Data Selection
Feature Selection techniques
Feature Weighting techniques
Clustering techniques
k-means method
g-means method
Nearest neighbour method
Association Models
Association Rules: A-priori algorithm
Discriminative/classification models
Decision trees induction: ID3 algorithm, C4.5 algorithm, CART algorithm
Classification rues induction: RULES algorithm, PRISM algorithm, CN2 algorithm, RISE algorithm
k-NN classifier
Predictive models
ANN
k-NN predictor
Rule-Based Reasoning module (rule inference engine o expert system shell)
Case-Based Reasoning module (case-based reasoning shell)

Most programming languages have available libraries for data science, for example *scikit-learn* for python or *JDMP* (*Java Data Mining Package*) for Java. Although they are not designed for graphical programming, the Flow Based Programming (FBP) paradigm described in Morrison, 2010) allows the programmer to create applications as a set of black boxes or interconnected processes. Nowadays many implementations of FBP can be found, such as NoFlo, NodeRed, or ThingsBoard, which are very much oriented to import and manipulate huge amounts of data.

On the other hand, there are some programming environments and languages that make the development process easier because are oriented to graphical programming, e.g., Matlab/Simulink (Champman, Stephen J., 2020) or LabVIEW (Johnson and Jennings, 2006), or further open-source equivalent options like Scilab (Campbell et al., 2010. Nagar, S., 2017) and MyOpenLab (Ruiz Gutierrez, J. M, 2017) respectively. These environments also have available data science libraries, as well as other specialized useful tools, for example database connection and reading data or data acquisition, among others, so they would be a better choice for implementation. Another advantage of these graphical environments is that they can be complemented with libraries from other programming languages —like C or Java— by creating new user defined blocks or tools, or using developed ones, e.g., Drools (Salantino, M., et al., 2016), a rules inference engine, developed in Java.

At the current stage, the software used for prototyping the methodology and the tool presented here is MATLAB-Simulink. This software provides all the necessary tools, a fourth-generation programming language (4GL) and a graphical programming environment that facilitates the standardization, allowing the tool to be easily reused in different installations.

The following subsections present the tool resulting from this work. All the developed methodologies have been integrated in an IDSS tool with the aim of being deployed in real facilities. MATLAB-Simulink has been used to implement all the methods presented in the previous chapters. Below are detailed all the implementations following

7.2.2 Data acquisition and storage

Data from the process is acquired through an OPC Server using the driver of the corresponding PLC in the facility, in a similar way as it is done by the SCADA system (Figure 7.7). Considering the problem described in Subsection 7.1.4, acquired data is stored in a local and dedicated database for the IDSS. A PostgreSQL open-source database (Hans-Jürgen, 2018) is used because of, first, it is open-source, and second, it is in line with the company roadmap regarding the standardisation and migration of some of their databases to PostgreSQL.

The PostgreSQL database contains a table to store the data from all variables, including those from sensors, other derived variables from them or set-points. Each row of the table corresponds to a sample, with three columns: variable name, value and sample time.

Furthermore, the database includes other complementary tables with the configuration and models required for the IDSS operation, such as the CB, the set of rules, parameters or KPI formulas.

7.2.3 Data science layer

The data science layer aim is to obtain reliable models from data. To this end, the data validation process illustrated in Figure 6.2 has been implemented using MATLAB-Simulink. The same tests proposed to validate online data can be applied to available historical data in order to use only good quality data to discover any kind of data-based model, mainly CB.

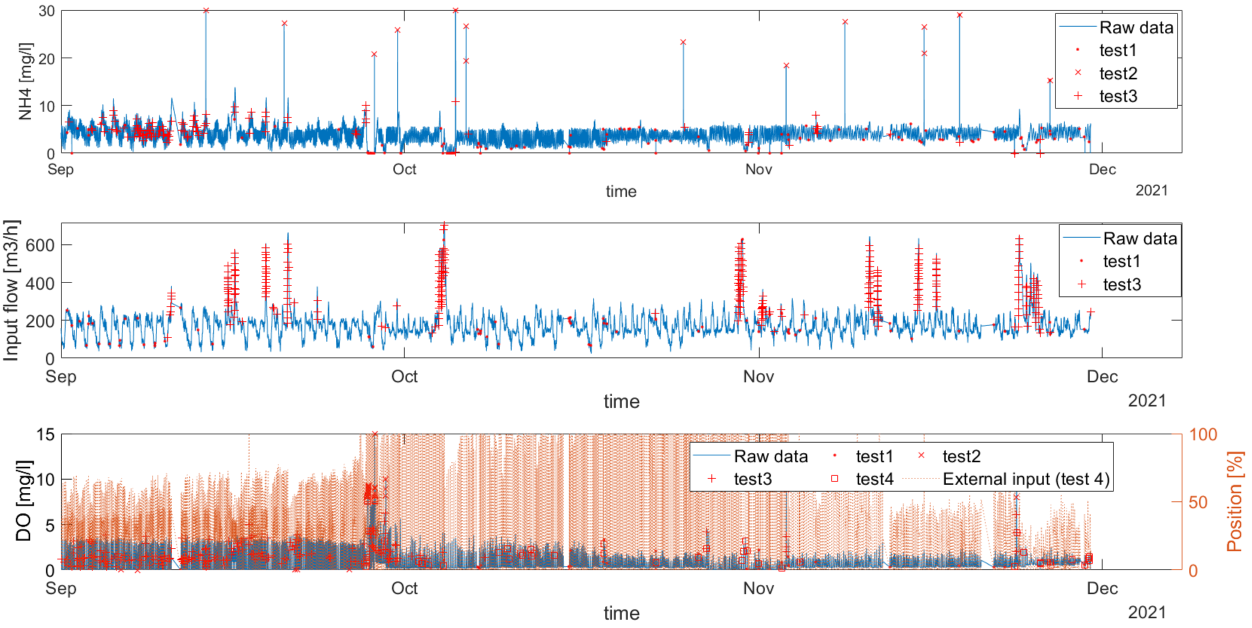


Figure 7.10 Identification of invalid values in a raw data set

Figure 7.10 illustrates the application of validation tests to 3 times-series data set. Table 7.6 details the rate of samples invalidated at each test for each time-series.

Table 7.6 Percentage of dismissed samples in time-series from Figure 7.10.

Percentage of invalidated samples				
Time-series	Test 1	Test 2	Test 3	Test 4
Ammonium (NH_4)	1,17	0,10	0,46	
Input flow	0,27	0	1,26	
Dissolved oxygen (DO)	0,16	0,19	0,82	0,23

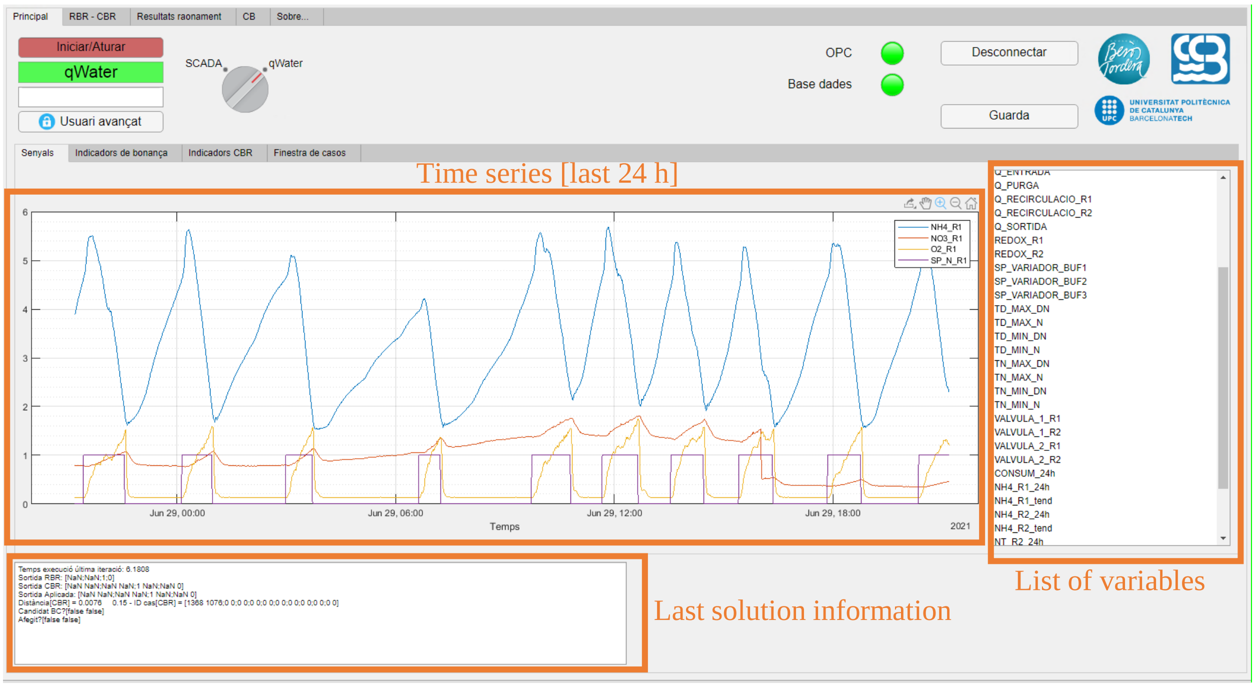
When applying validation tests to historical data, cases with one or more invalidated values are dismissed. Only in the online implementation of this method the imputation is considered. Results of the online validation and imputation process are presented in Section 7.4.

At this point, and in the light of the obtained results, it is worth highlighting the importance of the parametrization of the validation stage. Sometimes, the parameters could be completely objective, i.e., a negative concentration or a value above the upper limit of the sensor range. Other times, limits should be calculated statistically or considering the expert knowledge, i.e., the magnitude of the change in a concentration between two consecutive samples. Thus, it is necessary to be careful to avoid the invalidation of correct data, but also the validation of incorrect data. This issue becomes more important in the online implementation of the method. In Figure 7.10 some of these situations are illustrated. In the second time-series, which corresponds to the input flow, there are many values dismissed in test 3 (Trend test). The problem here is that the trend threshold has been calculated statistically. However, most of these invalidated values coincide in time with rainy episodes. This suggest that changes between samples in rainy episodes could be higher than the ones in dry episodes. This issue could be addressed by using a different parametrization of the validation module depending on the process situation, or considering, an external input, for example, data from a rain gauge, to modulate the trend threshold. Something similar is happening in the third time-series, the DO concentration. It can be observed a higher concentration of invalidated values in September, in comparison with October and November. But again, it can be observed a different behaviour between September and October-November. In this case, the causes may be quite different, for example, different boundary conditions (influent characterization), different operation conditions (the use of a powerful blower) or a real problem with the sensor.

7.2.4 Process design layer

As described in Subsection 3.2.2, the main purposes of this layer are the tool configuration, as well as being its GUI. The visual configuration capability illustrated in Figure 3.4 to connect different models with processes to be controlled and supervised has not been completely achieved in the framework of this thesis, but the proposed concept has inspired the methodologies implemented to date. On the other hand, a GUI has been developed to interact with the algorithms and methods provided by the IDSS, and to show to the user the status of the process through the time-series of measured variables and the configured KPIs.

Figure 7.11 is a screenshot of the main panel of the IDSS. From this screen the user can view the last 24 hours of the time-series of all measured variables, as well as the result of the last iteration of the reasoning process.

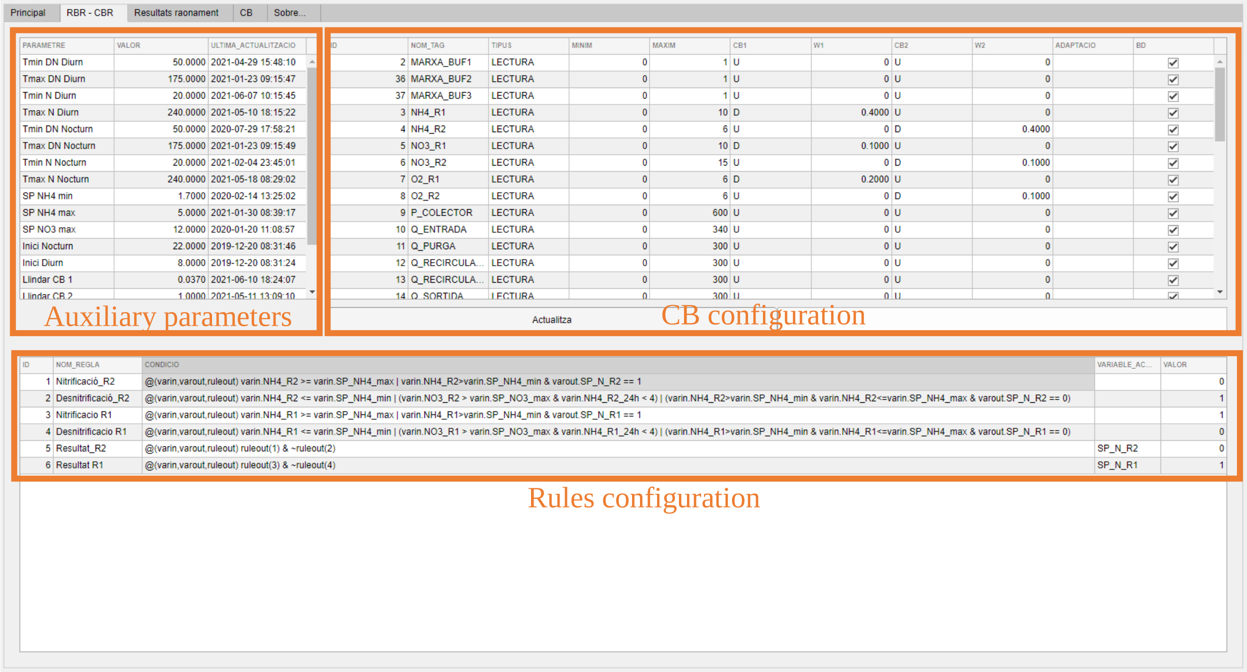


Q_ENTRADA
Q_PURGA
Q_RECIRCULACIO_R1
Q_RECIRCULACIO_R2
Q_SORTIDA
REDOX_R1
REDOX_R2
SP_VARIADOR_BUF1
SP_VARIADOR_BUF2
SP_VARIADOR_BUF3
TD_MAX_DN
TD_MAX_N
TD_MIN_DN
TD_MIN_N
TN_MAX_DN
TN_MAX_N
TN_MIN_DN
TN_MIN_N
VALVULA_1_R1
VALVULA_1_R2
VALVULA_2_R1
VALVULA_2_R2
CONSUM_24h
NH4_R1_24h
NH4_R1_tend
NH4_R2_24h
NH4_R2_tend
NT_R2_24h

```
Temps execució última iteració: 6.1808
Sortida RBR: [NaN;NaN;1;0]
Sortida CBR: [NaN NaN;NaN NaN;1 NaN;NaN 0]
Sortida Aplicada: [NaN NaN;NaN NaN;1 NaN;NaN 0]
Distància[CBR] = 0.0076     0.15 - ID cas[CBR] = [1368 1076;0 0;0 0;0 0;0 0;0 0;0 0;0 0;0 0]
Candidat BC?[false false]
Afegit?[false false]
```

Figure 7.11 Main screen of the IDSS (relevant parts)

From the screen in Figure 7.12 the CBR and RBR modules can be set. For the CBR module, the user can define the structure of the case, i.e., which features and solutions should be considered. For the RBR module, expert rules can be checked, and even modified. All this expert rule can use any available measured variable, as well as other constant parameters than can be also modified from this panel, i.e., the range of a set-point or the distance threshold considered to consider the current case a candidate case to be retained.



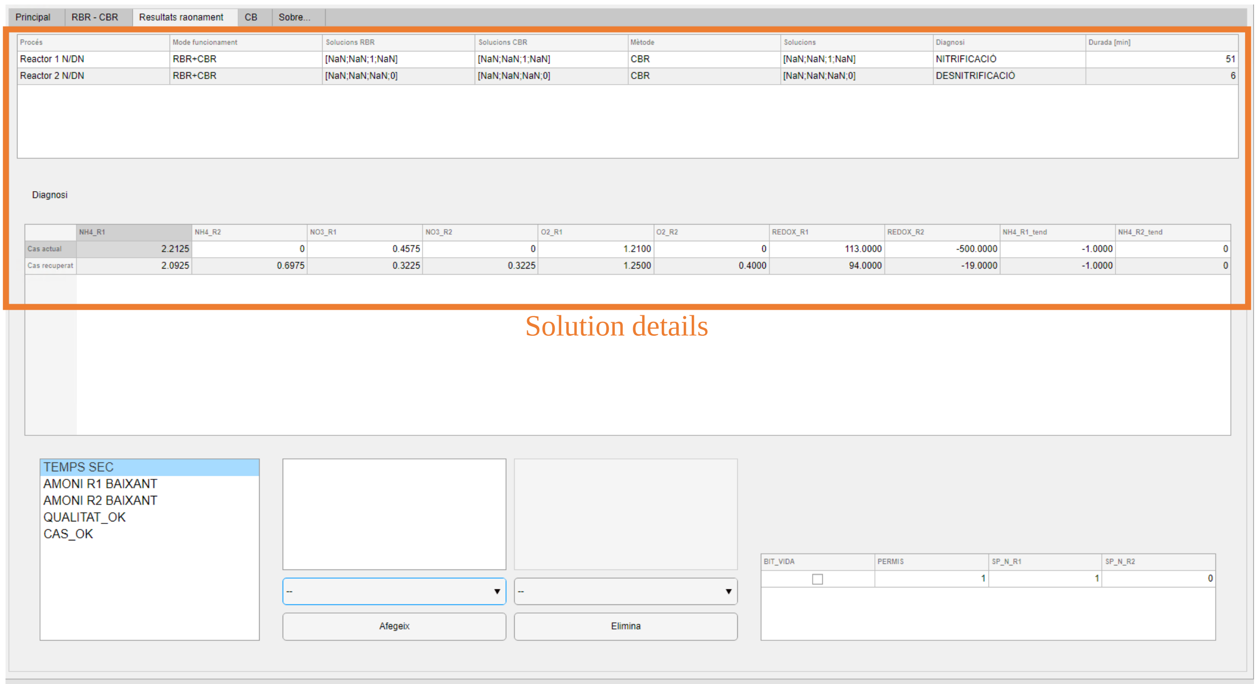
PARAMETRE	VALOR	ULTIMA_ACTUALITZACIO
Tmin DN Diurn	50.0000	2021-04-29 15:48:10
Tmax DN Diurn	175.0000	2021-01-23 09:15:47
Tmin N Diurn	20.0000	2021-06-07 10:15:45
Tmax N Diurn	240.0000	2021-05-10 18:15:22
Tmin DN Nocturn	50.0000	2020-07-29 17:58:21
Tmax DN Nocturn	175.0000	2021-01-23 09:15:49
Tmin N Nocturn	20.0000	2021-02-04 23:45:01
Tmax N Nocturn	240.0000	2021-05-18 08:29:02
SP NH4 min	1.7000	2020-02-14 13:25:02
SP NH4 max	5.0000	2021-01-30 08:39:17
SP NO3 max	12.0000	2020-01-20 11:08:57
Inici Nocturn	22.0000	2019-12-20 08:31:46
Inici Diurn	8.0000	2019-12-20 08:31:24
Llindar CB 1	0.0370	2021-06-10 18:24:07
Llindar CB 2	1.0000	2021-05-11 13:09:10

ID	NOM_TAG	TIPUS	MINIM	MAXIM	CB1	W1	CB2	W2	ADAPTACIO	BD
2	MARXA_BUF1	LECTURA	0	1	U	0	U	0		☑
36	MARXA_BUF2	LECTURA	0	1	U	0	U	0		☑
37	MARXA_BUF3	LECTURA	0	1	U	0	U	0		☑
3	NH4_R1	LECTURA	0	10	D	0.4000	U	0		☑
4	NH4_R2	LECTURA	0	6	U	0	D	0.4000		☑
5	NO3_R1	LECTURA	0	10	D	0.1000	U	0		☑
6	NO3_R2	LECTURA	0	15	U	0	D	0.1000		☑
7	O2_R1	LECTURA	0	6	D	0.2000	U	0		☑
8	O2_R2	LECTURA	0	6	U	0	D	0.1000		☑
9	P_COLECTOR	LECTURA	0	600	U	0	U	0		☑
10	Q_ENTRADA	LECTURA	0	340	U	0	U	0		☑
11	Q_PURGA	LECTURA	0	300	U	0	U	0		☑
12	Q_RECIRCULA...	LECTURA	0	300	U	0	U	0		☑
13	Q_RECIRCULA...	LECTURA	0	300	U	0	U	0		☑
14	Q_SORTIDA	LECTURA	0	300	U	0	U	0		☑

ID	NOM_REGLA	CONDICIO	VARIABLE_AC...	VALOR
1	Nitrificació_R2	@(varin,varout,ruleout) varin.NH4_R2 >= varin.SP_NH4_max | varin.NH4_R2>varin.SP_NH4_min & varout.SP_N_R2 == 1		0
2	Desnitrificació_R2	@(varin,varout,ruleout) varin.NH4_R2 <= varin.SP_NH4_min | (varin.NO3_R2 > varin.SP_NO3_max & varin.NH4_R2_24h < 4) | (varin.NH4_R2>varin.SP_NH4_min & varin.NH4_R2<=varin.SP_NH4_max & varout.SP_N_R2 == 0)		1
3	Nitrificacio R1	@(varin,varout,ruleout) varin.NH4_R1 >= varin.SP_NH4_max | varin.NH4_R1>varin.SP_NH4_min & varout.SP_N_R1 == 1		1
4	Desnitrificacio R1	@(varin,varout,ruleout) varin.NH4_R1 <= varin.SP_NH4_min | (varin.NO3_R1 > varin.SP_NO3_max & varin.NH4_R1_24h < 4) | (varin.NH4_R1>varin.SP_NH4_min & varin.NH4_R1<=varin.SP_NH4_max & varout.SP_N_R1 == 0)		0
5	Resultat_R2	@(varin,varout,ruleout) ruleout(1) & ~ruleout(2)	SP_N_R2	0
6	Resultat R1	@(varin,varout,ruleout) ruleout(3) & ~ruleout(4)	SP_N_R1	1

Figure 7.12 Configuration panels of the RBR and CBR modules in the IDSS (relevant parts)

Figure 7.13 screen provides to the user information about the solution obtained in the last iteration, how it has been obtained, i.e., RBR or CBR modules, and the similarity between the last case and the retrieved one.



Procés	Mode funcionament	Solucions RBR	Solucions CBR	Mètode	Solucions	Diagnosi	Durada [min]
Reactor 1 N/DN	RBR+CBR	[NaN;NaN;1;NaN]	[NaN;NaN;1;NaN]	CBR	[NaN;NaN;1;NaN]	NITRIFICACIÓ	51
Reactor 2 N/DN	RBR+CBR	[NaN;NaN;NaN;0]	[NaN;NaN;NaN;0]	CBR	[NaN;NaN;NaN;0]	DESNITRIFICACIÓ	6

Diagnosi

	NH4_R1	NH4_R2	NO3_R1	NO3_R2	O2_R1	O2_R2	REDOX_R1	REDOX_R2	NH4_R1_tend	NH4_R2_tend
Cas actual	2.2125	0	0.4575	0	1.2100	0	113.0000	-500.0000	-1.0000	0
Cas recuperat	2.0925	0.6975	0.3225	0.3225	1.2500	0.4000	94.0000	-19.0000	-1.0000	0

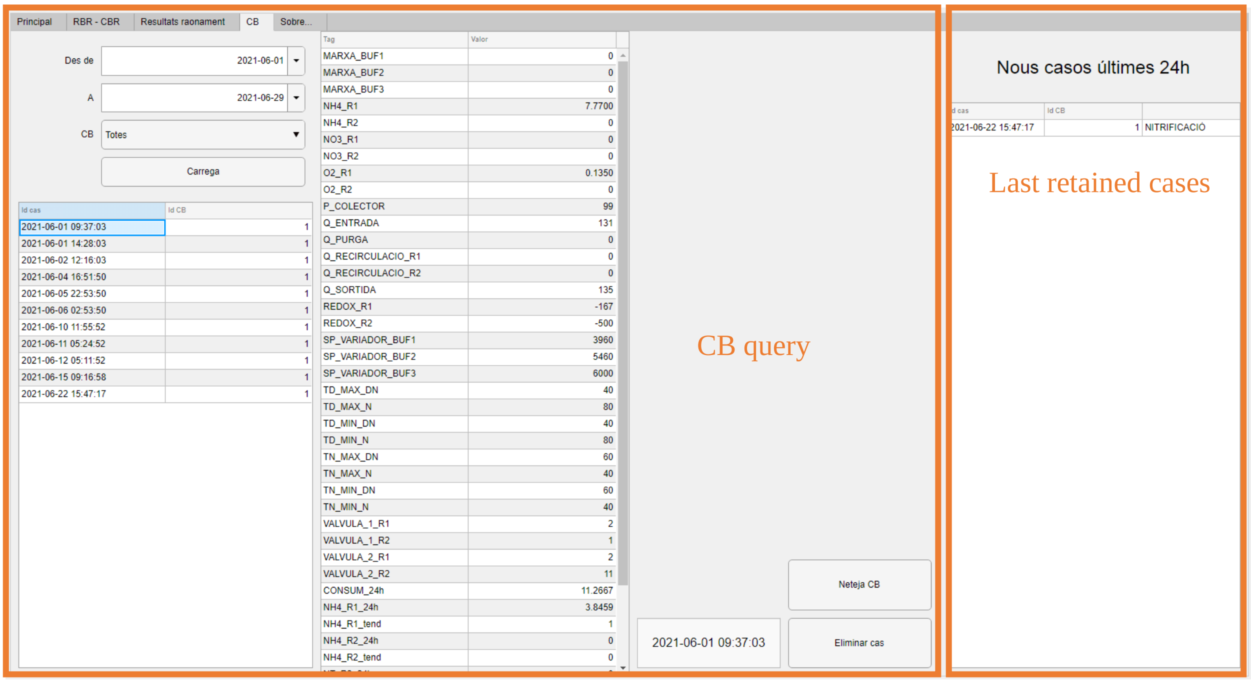
Solution details

TEMPS SEC
AMONI R1 BAIXANT
AMONI R2 BAIXANT
QUALITAT_OK
CAS_OK

BIT_VIDA	PERMIS	SP_N_R1	SP_N_R2
□	1	1	0

Figure 7.13 Solutions panel in the IDSS (relevant parts)

In Figure 7.14 screen the user can check the case base, filtering the retained cases between two dates, and displaying the values for all features and solutions. In the right-hand side panel cases retained in the last 24 hours are listed.



Des de 2021-06-01
A 2021-06-29
CB Totes

Id cas	Id CB
2021-06-01 09:37:03	1
2021-06-01 14:28:03	1
2021-06-02 12:16:03	1
2021-06-04 16:51:50	1
2021-06-05 22:53:50	1
2021-06-06 02:53:50	1
2021-06-10 11:55:52	1
2021-06-11 05:24:52	1
2021-06-12 05:11:52	1
2021-06-15 09:16:58	1
2021-06-22 15:47:17	1

Tag	Valor
MARXA_BUF1	0
MARXA_BUF2	0
MARXA_BUF3	0
NH4_R1	7.7700
NH4_R2	0
NO3_R1	0
NO3_R2	0
O2_R1	0.1350
O2_R2	0
P_COLECTOR	99
Q_ENTRADA	131
Q_PURGA	0
Q_RECIRCULACIO_R1	0
Q_RECIRCULACIO_R2	0
Q_SORTIDA	135
REDOX_R1	-167
REDOX_R2	-500
SP_VARIADOR_BUF1	3960
SP_VARIADOR_BUF2	5460
SP_VARIADOR_BUF3	6000
TD_MAX_DN	40
TD_MAX_N	80
TD_MIN_DN	40
TD_MIN_N	80
TN_MAX_DN	60
TN_MAX_N	40
TN_MIN_DN	60
TN_MIN_N	40
VALVULA_1_R1	2
VALVULA_1_R2	1
VALVULA_2_R1	2
VALVULA_2_R2	11
CONSUM_24h	11.2667
NH4_R1_24h	3.8459
NH4_R1_tend	1
NH4_R2_24h	0
NH4_R2_tend	0

CB query

2021-06-01 09:37:03

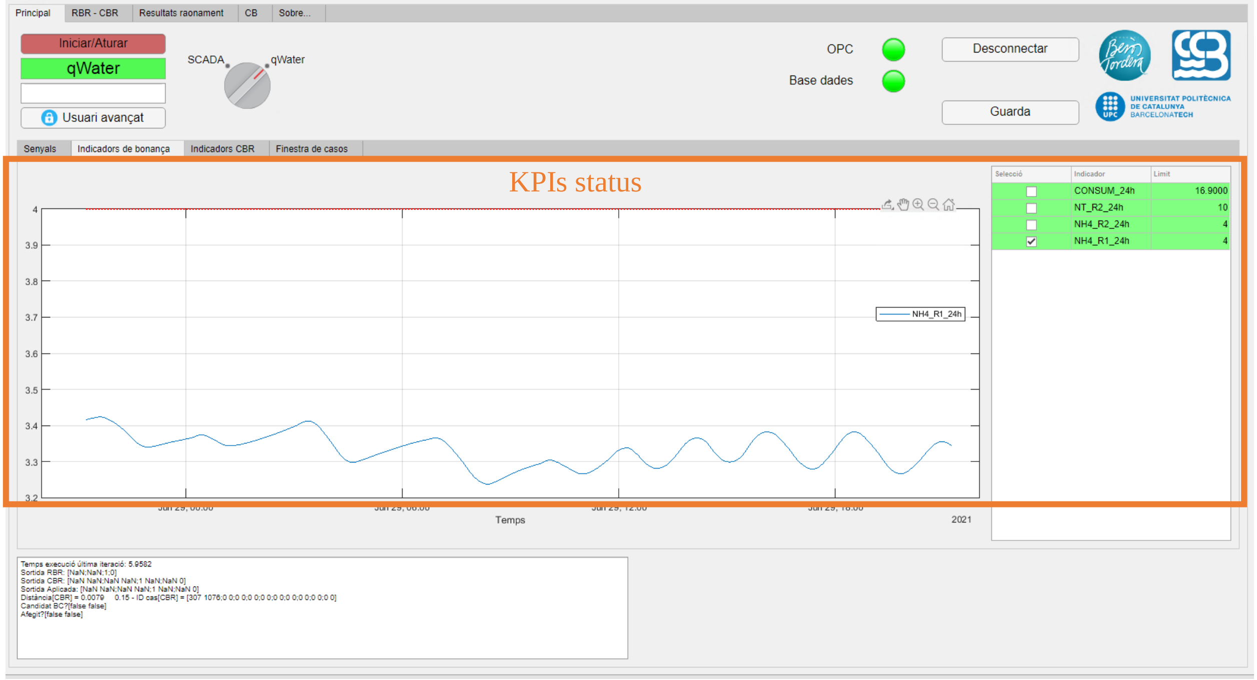
Nous casos últimes 24h

d cas	Id CB	
2021-06-22 15:47:17	1	NITRIFICACIÓ

Last retained cases

Figure 7.14 CB panel in the IDSS (relevant parts)

In Figure 7.15 screen the user can check the status of the configured KPIs using a visual traffic light-based table (informing for good – green, acceptable – orange or bad – red status), as well as the last 24 hours evolution.

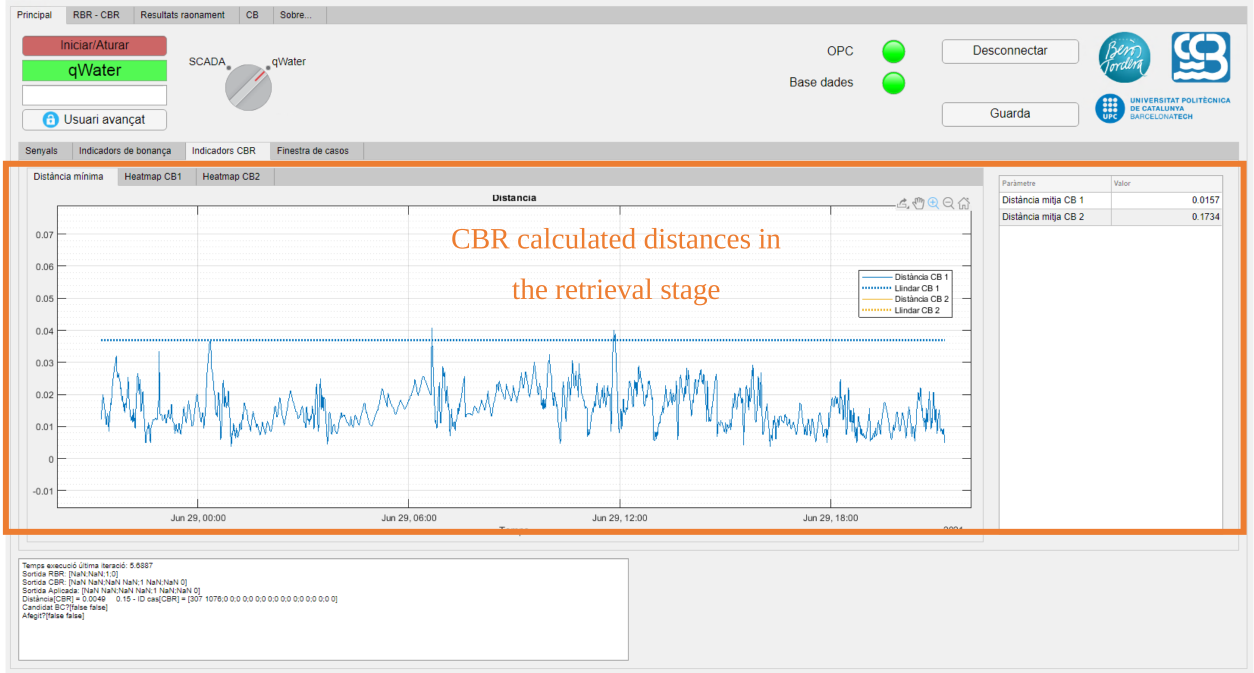


Selecció	Indicador	Limit
□	CONSUM_24h	16.9000
□	NT_R2_24h	10
□	NH4_R2_24h	4
☑	NH4_R1_24h	4

```
Temps execució última iteració: 5.9582
Sortida RBR: [NaN;NaN;1;0]
Sortida CBR: [NaN NaN;NaN NaN;1 NaN;NaN 0]
Sortida Aplicada: [NaN NaN;NaN NaN;1 NaN;NaN 0]
Distància[CBR] = 0.0079     0.15 - ID cas[CBR] = [307 1076;0 0;0 0;0 0;0 0;0 0;0 0;0 0;0 0;0 0]
Candidat BC?[false false]
Afegit?[false false]
```

Figure 7.15 KPIs panel in the IDSS (relevant parts)

In Figure 7.16 panel the user can see the time-series of the evolution of the distance between received cases and the retrieved ones, and how many of the have been candidates to be retained.

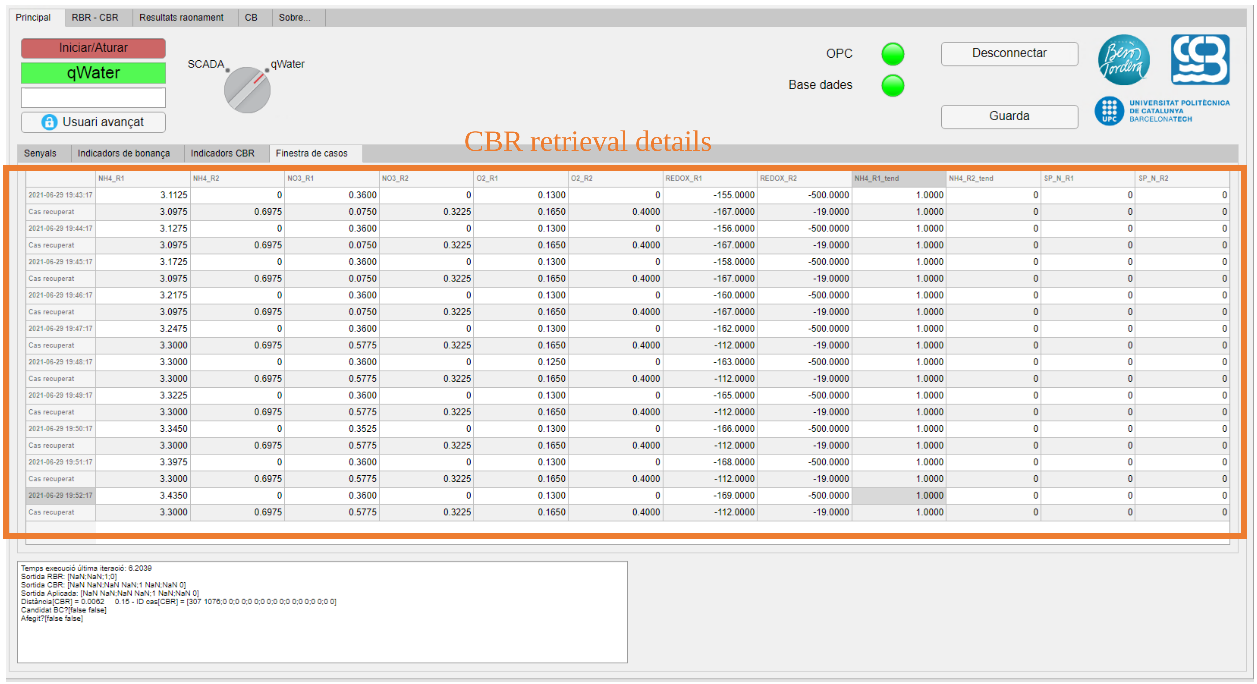


Paràmetre	Valor
Distància mitja CB 1	0.0157
Distància mitja CB 2	0.1734

```
Temps execució última iteració: 5.6887
Sortida RBR: [NaN;NaN;1;0]
Sortida CBR: [NaN NaN;NaN NaN;1 NaN;NaN 0]
Sortida Aplicada: [NaN NaN;NaN NaN;1 NaN;NaN 0]
Distància[CBR] = 0.0049     0.15 - ID cas[CBR] = [307 1076;0 0;0 0;0 0;0 0;0 0;0 0;0 0;0 0;0 0]
Candidat BC?[false false]
Afegit?[false false]
```

Figure 7.16 CBR performance panel in the IDSS (relevant parts)

From the panel in Figure 7.17 the user can compare the real cases around a particular point of time with the retrieved ones.



	NH4_R1	NH4_R2	NO3_R1	NO3_R2	O2_R1	O2_R2	REDOX_R1	REDOX_R2	NH4_R1_tend	NH4_R2_tend	SP_N_R1	SP_N_R2
2021-06-29 19:43:17	3.1125	0	0.3600	0	0.1300	0	-155.0000	-500.0000	1.0000	0	0	0
Cas recuperat	3.0975	0.6975	0.0750	0.3225	0.1650	0.4000	-167.0000	-19.0000	1.0000	0	0	0
2021-06-29 19:44:17	3.1275	0	0.3600	0	0.1300	0	-156.0000	-500.0000	1.0000	0	0	0
Cas recuperat	3.0975	0.6975	0.0750	0.3225	0.1650	0.4000	-167.0000	-19.0000	1.0000	0	0	0
2021-06-29 19:45:17	3.1725	0	0.3600	0	0.1300	0	-158.0000	-500.0000	1.0000	0	0	0
Cas recuperat	3.0975	0.6975	0.0750	0.3225	0.1650	0.4000	-167.0000	-19.0000	1.0000	0	0	0
2021-06-29 19:46:17	3.2175	0	0.3600	0	0.1300	0	-160.0000	-500.0000	1.0000	0	0	0
Cas recuperat	3.0975	0.6975	0.0750	0.3225	0.1650	0.4000	-167.0000	-19.0000	1.0000	0	0	0
2021-06-29 19:47:17	3.2475	0	0.3600	0	0.1300	0	-162.0000	-500.0000	1.0000	0	0	0
Cas recuperat	3.3000	0.6975	0.5775	0.3225	0.1650	0.4000	-112.0000	-19.0000	1.0000	0	0	0
2021-06-29 19:48:17	3.3000	0	0.3600	0	0.1250	0	-163.0000	-500.0000	1.0000	0	0	0
Cas recuperat	3.3000	0.6975	0.5775	0.3225	0.1650	0.4000	-112.0000	-19.0000	1.0000	0	0	0
2021-06-29 19:49:17	3.3225	0	0.3600	0	0.1300	0	-165.0000	-500.0000	1.0000	0	0	0
Cas recuperat	3.3000	0.6975	0.5775	0.3225	0.1650	0.4000	-112.0000	-19.0000	1.0000	0	0	0
2021-06-29 19:50:17	3.3450	0	0.3525	0	0.1300	0	-166.0000	-500.0000	1.0000	0	0	0
Cas recuperat	3.3000	0.6975	0.5775	0.3225	0.1650	0.4000	-112.0000	-19.0000	1.0000	0	0	0
2021-06-29 19:51:17	3.3975	0	0.3600	0	0.1300	0	-168.0000	-500.0000	1.0000	0	0	0
Cas recuperat	3.3000	0.6975	0.5775	0.3225	0.1650	0.4000	-112.0000	-19.0000	1.0000	0	0	0
2021-06-29 19:52:17	3.4350	0	0.3600	0	0.1300	0	-169.0000	-500.0000	1.0000	0	0	0
Cas recuperat	3.3000	0.6975	0.5775	0.3225	0.1650	0.4000	-112.0000	-19.0000	1.0000	0	0	0

```
Temps execució última iteració: 6.2039
Sortida RBR: [NaN;NaN;1;0]
Sortida CBR: [NaN NaN;NaN NaN;1 NaN;NaN 0]
Sortida Aplicada: [NaN NaN;NaN NaN;1 NaN;NaN 0]
Distància[CBR] = 0.0062     0.15 - ID cas[CBR] = [307 1076;0 0;0 0;0 0;0 0;0 0;0 0;0 0;0 0;0 0]
Candidat BC?[false false]
Afegit?[false false]
```

Figure 7.17 Case retrieval panel in the IDSS (relevant parts)

7.2.5 Process control layer

The process control layer is based on the methodology detailed in Chapters 4 and 5. The reasoning scheme in Figure 4.1 has been implemented using MATLAB-Simulink software, mainly using the Simulink tool (Figure 7.18), supported by MATLAB functions.

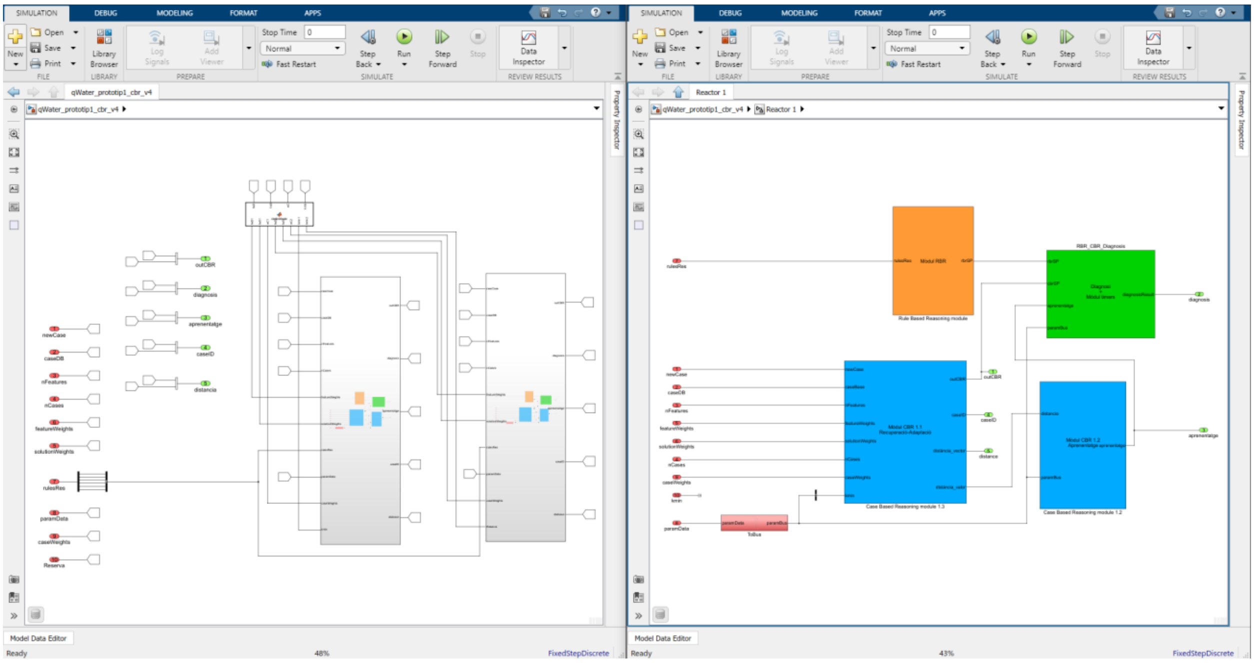


Figure 7.18 Implementation of the Process control layer using the Simulink environment

The concept proposed in Chapter 3 facilitates the development of a scalable system, with the idea of being deployed in different real installations with minor changes. In general, differences between installations can be simplified in differences in the input data to the process control workflow in Figure 3.5, but the CBR and RBR solutions are the same for all of them. It means that the source code of the IDSS is the same for any installation, in this case, for the two facilities presented in 7.1. The CB and rules feeding CBR and RBR modules respectively are not embedded in the code, but codified in the IDSS database, so reconfiguring the database is the only step to adapt the process control workflow to a particular installation. In short, to adapt the IDSS to a certain system, it is only necessary to know the domain data and build valid models.

7.3. IDSS operation in real facilities

The IDSS presented in the previous section has been tested in two installations: *Santa Maria de Palautordera* and *Castellar del Vallès* WWTPs, which are described in the Case study Chapter, in Subsections 7.1.2 and 7.1.3. The IDSS has intermittently been working in both installations since February 2020 and February 2021 respectively, accumulating more than one year of operation.

The most important parameters that should be configured for the operation of the tool in a real installation are:

- The distance threshold, which determines how different will be the retained cases in the CB.
- The case configuration, including which parameters should be considered and the weight assigned to each one.

All these can be modified by the user from the GUI. The initialization of the distance threshold can be done using the procedure explained in Section 4.3. The idea is to learn only relevant cases, i.e., cases which are enough different to those in the CB. Figure 7.19 illustrates this procedure. A set of cases obtained during one month is considered. The first fifteen days are used as an initial CB to calibrate the threshold. The minimum distance between each case and the others is calculated (blue dotted line in Figure 7.19). Then, using the average (4-7) and the standard deviation (4-8) of all distances the threshold (4-9) is calculated.

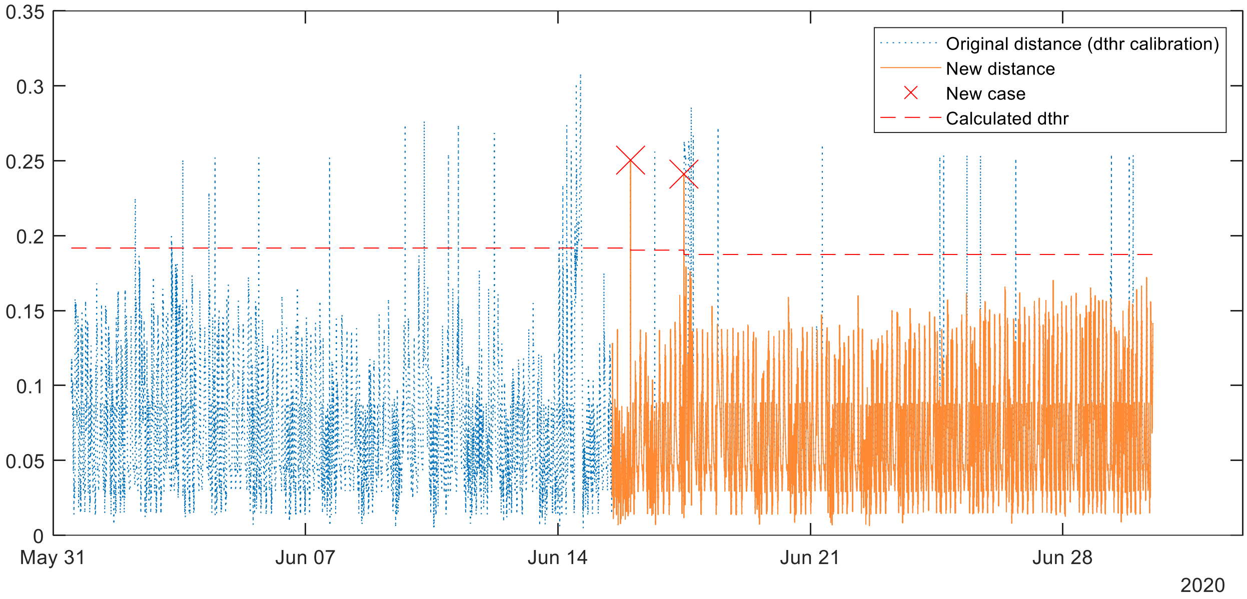


Figure 7.19 Example of distance threshold calculation using real data

From this point, and using the calculated threshold ($d_{thr} = 0.192$), each time a new case is received the most similar one is retrieved from the CB. If the distance is below the threshold, the case is not retained. Otherwise, if the distance between the current case and the most similar one (orange continuous line in Figure 7.19) in the CB is over the threshold, the case is a candidate to be retained. In this example, and assuming that the revision stage is positive, there are two learned cases (red cross in Figure 7.19). The threshold could be a static value, or recalculated each time a new case is retained.

The CB configuration is also an important step. Here, the user can decide which variables are used as features of the case, as well as the weight of each one, and which ones are solutions. By default, the weight of all features is the same. At the current version, the user can adjust the weights to different values, if considered necessary. In the future, an automatic calibration method could be integrated in the tool.

To evaluate the performance of the tool some KPIs have been defined, in collaboration with the plant managers. These indicators are considered adequate to supervise the performance of the biological process and are related to the outflow quality or the WWTP efficiency from the consumption point of view. In addition, they are standard indicators that can be used in any installation. These indicators are:

- 24h moving average (MA) of ammonium concentration (24MA-AC): The 24h MA of ammonium concentration in the effluent is stablished by applicable regulations to a maximum value of 4 mg/l.
- Blower electrical consumption (BEC): The daily average consumption is calculated with historical data and used as a threshold to be compared with the current daily average consumption.
- Total nitrogen concentration (TNC): Total nitrogen in the influent and in the effluent of the WWTP is not an online measure, but an offline analytic measure obtained three times per week. The total nitrogen concentration in the effluent is stablished by applicable regulations to a maximum value

of 10 mg/l. In terms of nitrogen removal, an efficiency around 80% is considered a good performance.

Ammonium and total nitrogen concentrations in the WWTP effluent are not online measurements. They are obtained two or three times per week from daily integrated samples collected with an automatic sampler (Figure 7.20) and analysing them in the laboratory. A daily integrated sample consists of a set of 24 subsamples, one per hour, and finally integrated in a unique sample. Usually, two or three subsamples are taken automatically every hour. To complement these offline indicators, ammonium sensors installed in the biological reactor are used as an approximation of the ammonium in the effluent, which is the value to be considered by the regulators. It is expected that the ammonium in the effluent is lower than the measured value in the biological reactors due to the nitrification occurred in the secondary decanters.

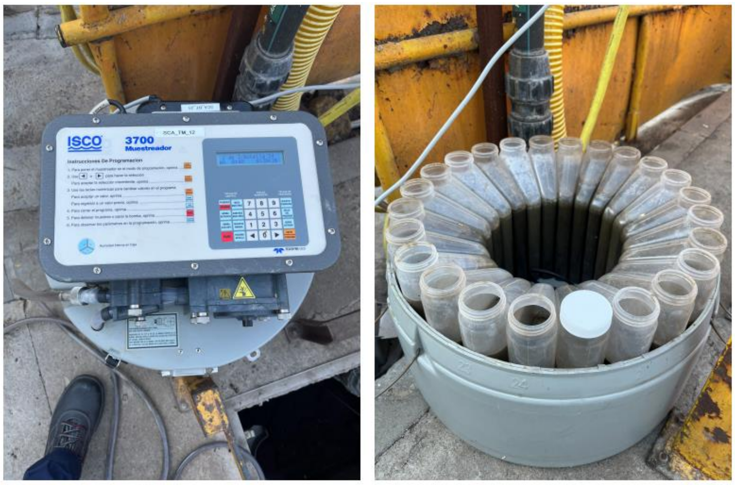


Figure 7.20 Automatic sampler: configuration panel (left) and interior of the sampler with 24 bottles for samples.

Regarding the consumption, most WWTP have network analysers registering many electrical parameters in real time, such as the consumption. The problem here is that consumptions are not usually separated by equipment, but for areas. So, in the case of the WWTP in the study, we cannot obtain the aeration process consumption disaggregated from the total consumption of a particular area. Instead of the consumption two alternatives can be considered: the total consumption or the consumption of the area where the blowers are connected, or the rate between aeration and absence of aeration. Of course, both options are an approximation, and it has to be assumed, first, that the total consumption can vary due to other equipment, and second, that the relation between the consumption and the aeration time is not necessarily linear.

Figure 7.21 shows the baseline for ammonium, nitrogen and electrical consumption. The two years previous to the IDSS deployment are considered to assess the impact of the IDSS in the process performance. Looking at Figure 7.21, it can be observed that in the case of ammonium and nitrogen, most samples are under the allowed limit, before and after the IDSS (hence the system is performing adequately both with and without the IDSS according to these KPIs). In the case of electrical consumption, the mean daily consumption increases negligibly from year to year, almost negligible.

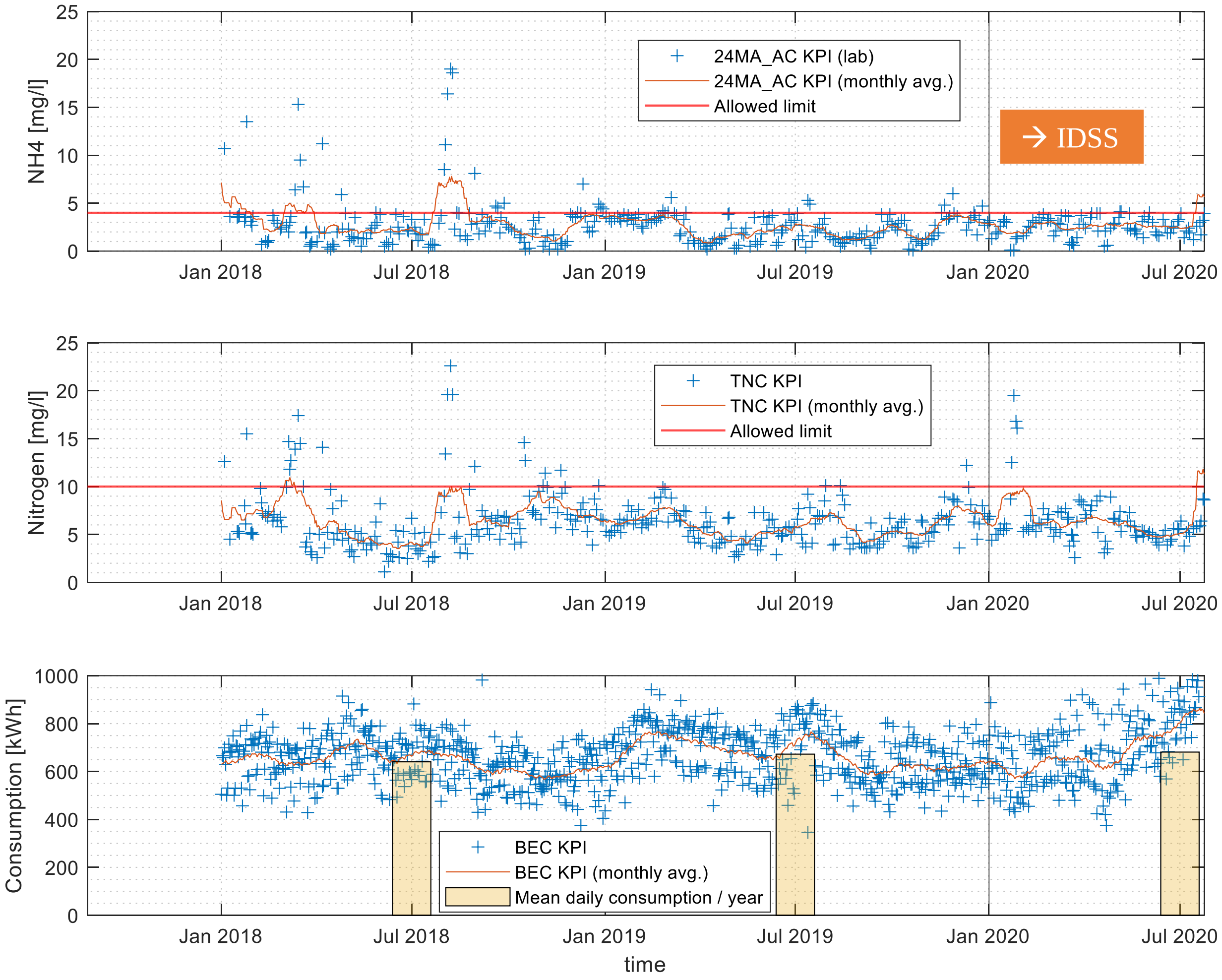


Figure 7.21 Ammonium, nitrogen and electrical consumption KPI baseline (before IDSS)

Figure 7.22, Figure 7.23 and Figure 7.24 show in more detail the evolution of the proposed KPIs during a period of 8 months after the deployment of the IDSS. The initial CB does not consider historical data. The problem detected here was with the quality of historical data, and also the difficulty to obtain it from the plant SCADA system. In addition, the plant equipment was improved with new sensors and the plant configuration was changed (from 1 to two biological reactors). For all these reasons, the system was initialized with some expert rules and an empty CB. After two months operation, the procedure described in Figure 7.19 was applied. With the help of the expert users, different types of cases where identified and the CB was simplified applying a clustering to the available data. In the dataset, four types of cases were identified, and 10 cases from each type were randomly selected as an initial CB.

In Figure 7.22, the 24h mean average calculated using online data from the NH4 sensor in the biological reactor (24MA-AC online) is compared with the analytical results obtained in the laboratory (24MA-AC offline) from samples collected in the effluent, where the limit of four mg/l has to be achieved. The effluent is analysed about three times per week. It can be observed that almost all analysed samples (blue +) are below the limit, i.e. the operation is correct. From February to mid-May it can be noted that the NH4 sensor measurements (orange x) are well above the limit due to a calibration problem in the sensor. Of course, cases during this period cannot be considered as possible candidates to be retained.

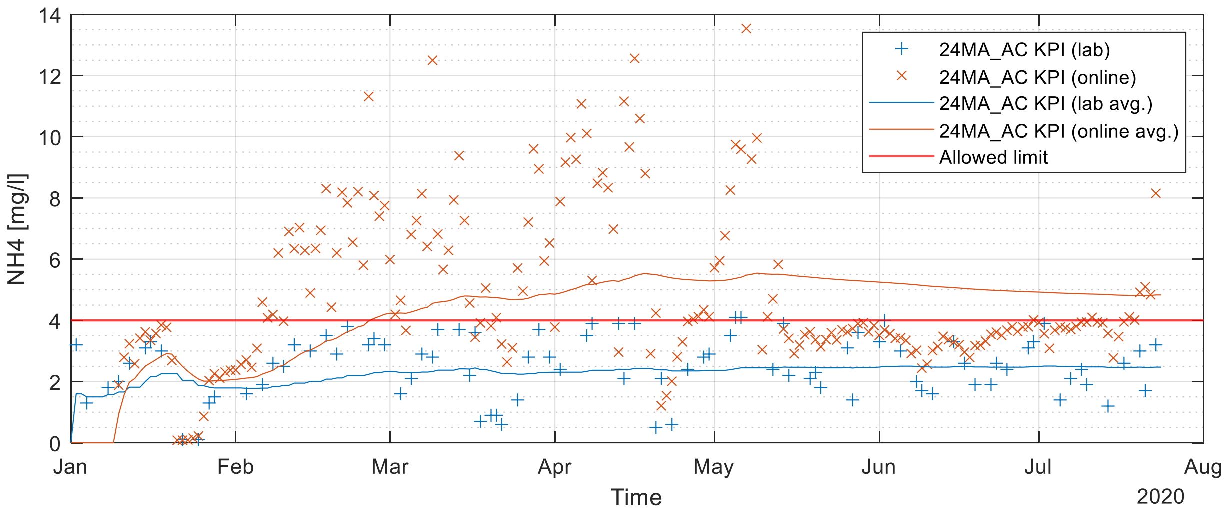


Figure 7.22 NH4 24h average (24MA-AC) KPI for the test period

Regardless the calibration problem, it can be observed that the difference between lab values and online measurements is surprisingly high in some cases, even reaching 2 mg/l or more. This difference can be explained for the sensors considered. For the particular case of the ammonium, the sensor is not placed in the same point where the sample is taken to be analysed in the lab. Ammonium sensors are installed at the end of the biological reactor and their measures are used for the process control. On the other hand, samples for lab analysis are taken in the effluent of the process, after the secondary clarifiers. So, this is the main reason of that difference. Another reason is due to precision of the sensors compared with lab analysis. To control the process, the precision provided by an ammonium sensor is considered sufficient. On the other hand, the ammonium concentration in the effluent is regulated and must be fulfilled, and this is why samples are obtained and analysed periodically. If more reliable values were required for control, ammonium analysers could be used instead of probes.

In Figure 7.23, the electrical consumption (BEC) indicator of the studied period is compared with the mean electrical consumption of the same period in 2019. Here, the total consumption is considered. In addition, the ratio of this electrical consumption related to the volume of treated water is shown. The electrical consumption is similar to the one before the deployment of the IDSS tool. The ratio between treated water and consumption is 0.35 kwh/m^3, below the 0.40 kwh/m^3 during the same period in 2019.

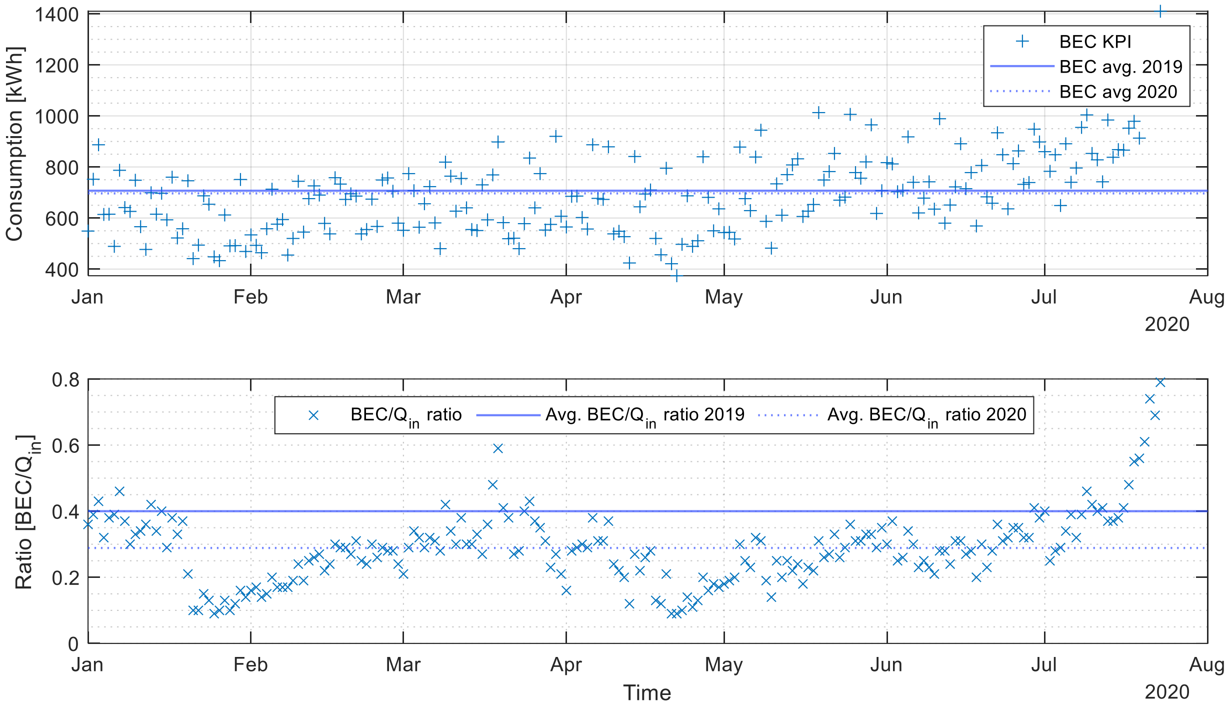


Figure 7.23 BEC KPI for the test period

Finally, Figure 7.24 shows the total nitrogen concentration (TNC) in the effluent (blue +) compared to the allowed limit (blue line). Some values over the limit can be observed in the last days of January 2020. These results are due to an intense rainy period that caused a high increase of the nitrate concentration in the WWTP influent, which is one of the components of the total nitrogen. The European Union directive 91/271/CEE on urban waste water treatment stablishes the maximum nitrogen concentration in the effluent or the minimum nitrogen removal efficiency (orange x) depending on the WWTP influent load, expressed in population equivalent (PE) units. In terms of nitrogen removal efficiency, it can be pointed out that the mean removal efficiency is about 80%. The European directive stablishes a minimum value between 70 and 80%. So, in terms of nitrogen removal results are also within the limits.

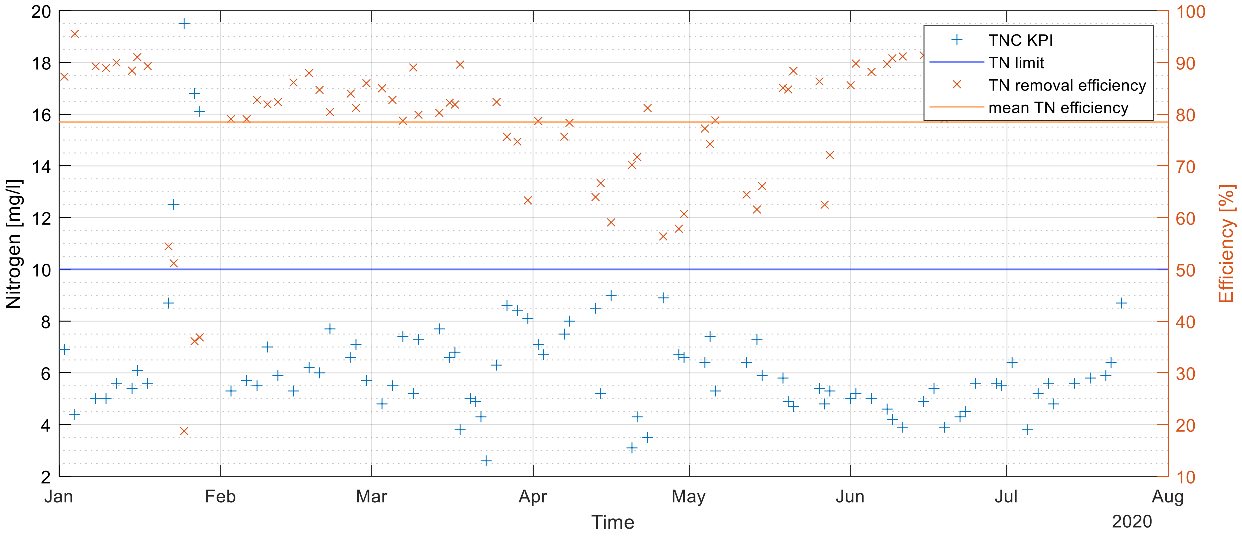


Figure 7.24 Total Nitrogen (TNC) KPI and Nitrogen removal efficiency for the test period

Performance indicators presented in the above figures are some of the common indexes used by plant managers and operators to assess how the process is working. The following indicators are proposed to quantify the competence of the system in the problem resolution:

- *Solved cases* (SC): Percentage of solved cases. This index indicates the period when the tool is operating. The tool may not be operating during, for example, maintenance tasks, either of the plant or the application itself. It could be a good index to quantify the adaptation of the system to the different situations occurred in the plant, i.e., low or high load influents, dry or rainy periods, etc.
- *CBR index* (CBR_i): Percentage of cases solved by CBR module. This index is useful to quantify the confidence and/or competence in CBR module.
- *RBR index* (RBR_i): Percentage of cases solved by RBR module. This index is useful to quantify the rate of situations that can’t be solved by the CBR module, i.e., candidate cases.
- *Expert index* (E_i): Percentage of cases that are not solved by the reasoning cycle., i.e. the proposed solution is not the one used due to maintenance tasks in the WWTP or open-loop fixed timers set-points.
- *Retain index* (R_i): Number of retained cases. This index considers the new learned cases to the original case base of 40 cases described above.
- *Correctly solved cases* (CSC*):* Percentage of correctly solved cases by the whole reasoning system – i.e. CBR and RBR – considering the expert assessment.

Table 7.7 shows the monthly values for these indexes, and the average value considering data from April to August. The first three months are used for the tool set-up.

Table 7.7 Competence of the reasoning system

Period	**SC [%]**	**CBR_i [%]**	**RBR_i [%]**	**E_i [%]**	**R_i**	**CSC [%]**
January°	71.0	0	100	nd	nd	nd
February°	99.5	0	100	nd	nd	nd
March°	88.5	nd	nd	16.0	nd	nd
April	94.7	100	0	15.6	1	85.04
May	84.6	99.98	0.02	9.2	9	92.00
June	99.3	100	0	4.4	0	96.56
July	71.4	99.92	0.08	9.1	24	91.08
August	97.2	99.94	0.06	7.4	25	94.01
Total#	89.4	99.97	0.03	9.1	59	91.74

nd: not determined

° results during this period are approximate values (tool set-up)

# period from April to August is considered

One of the challenges to face in the design of the IDSS is how to integrate the human operator in the decision-making process. The user can tune many parameters, such as feature weights, expert rules, etc. but also is integrated in the reasoning system of the tool, providing his feedback in the revision process. Ideally, the revision step should be automated, but the reality is that in complex processes, e.g., waste water treatment, the human operator plays a key role in the IDSS performance. As previously described, some situations, such as boundary conditions, e.g., a contaminant discharge, can trigger situations difficult to be detected automatically by available real time monitoring sensors, but maybe detected by a laboratory analysis or even by visual inspection. Thus, human operator's knowledge is crucial for the successful performance of the process. As described above, IDSS performance is evaluated using some familiar KPIs to the user. Therefore, KPI values are not only an indicator of how reliable are the rules or CB models used, but also evaluate how the user is tuning and providing feedback the IDSS.

Regarding the temporal approach proposed in Chapter 5, results are obtained using real offline data. In the first experiments, a one-year (2021) dataset is considered. The 70:30 ratio is used to split data into calibration and validation. Thus, the first 8,5 months are used to generate an EB model, validated with the 3,5 remaining months (subset 0 in Table 7.8). Then, 3 more one-month length validation subsets from 2022 are also used in order to check the robustness of the model (Subset 1, Subset 2 and Subset 3).

To determine the best episode length, i.e., the one maximizing the reasoning system performance, a tuning process with different episodes lengths is used. Figure 7.25 shows an example of a six cases-length episode.

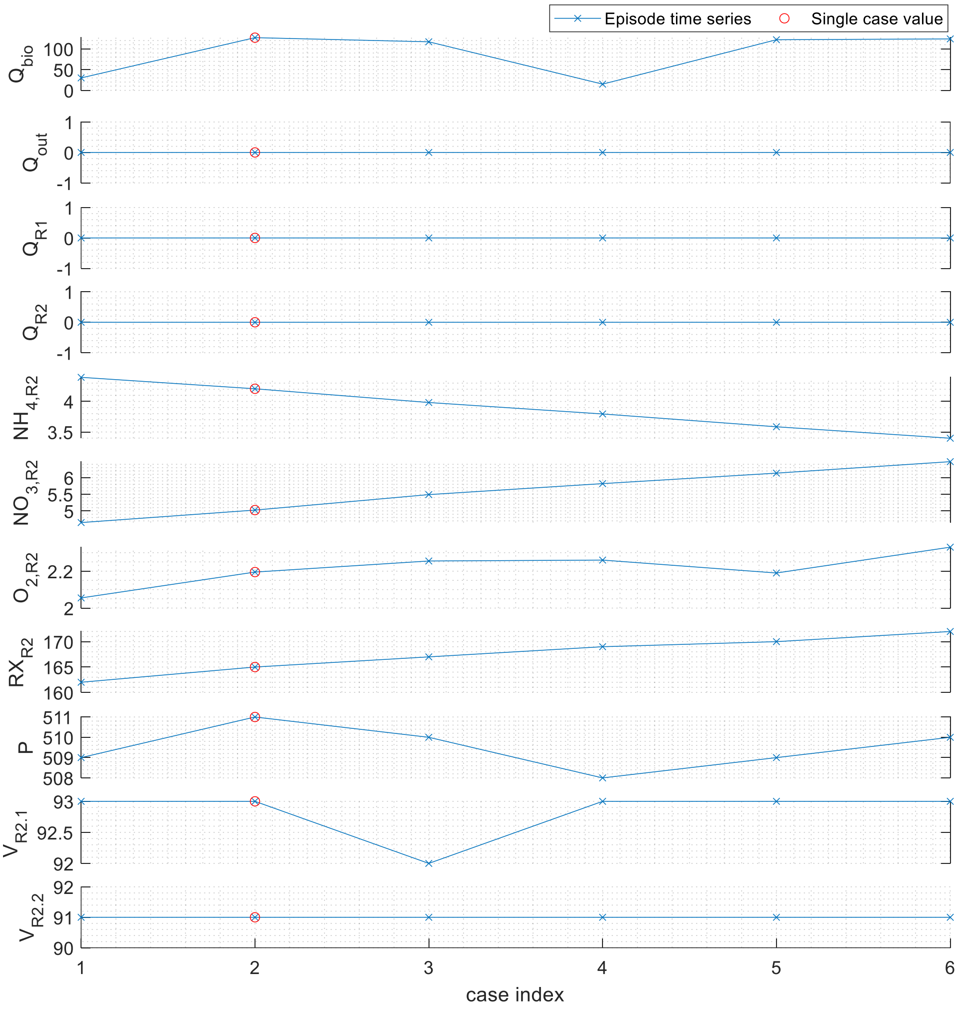


Figure 7.25 Example of a 6 samples length episode

The performance of the temporal approach is evaluated calculating the accuracy of the proposed solution when using the temporal CBR with respect to the actual or expected one. Thus, if the reused solution, i.e., set of set-points, from the most similar episode is the one expected, the solution is considered good, i.e., accuracy is 100%. Otherwise, accuracy is 0%. In addition, two additional models are calibrated to be compared with the case-based reasoning approach, an ANN and a LR model.

The method performance is not the only factor to consider, also computing time and EB size are important issues to take into consideration for the selection of the episode length. Episodes' length from 1 case, i.e., classic CBR to 24 cases, i.e., 2 hours, since the sample time is 5 minutes, are explored. Despite different dynamics can be found in wastewater treatment process, considering the involved and measured variables in the process described in the case study, the range from 1 to 24 cases is considered appropriate to capture the behaviour of the process. Of course, for processes or domains with variables with different dynamics the use of fixed or variable length episodes should be explored in order to find the best solution for that particular cases.

Figure 7.26 and Table 7.8 show the accuracy obtained depending on the episode's length in the EB, for the case of TCBR method, as well as the accuracy using an ANN model and LR model.

As it can be seen in Figure 7.26, on the left side, the accuracy increases when using episodes with length 2 or more cases. From a specific number of cases accuracy value remains stable. Results are shown for the training subset (dashed line), i.e., the one used to generate the EB, and the mean accuracy calculated from other 3 subsets (solid line). The accuracy interval considering the three validation subsets is also represented. On the right side of Figure 7.26, the accuracy obtained using two additional models, – i.e. an ANN and a LR model–, is represented to be compared with the one obtained with the TCBR.

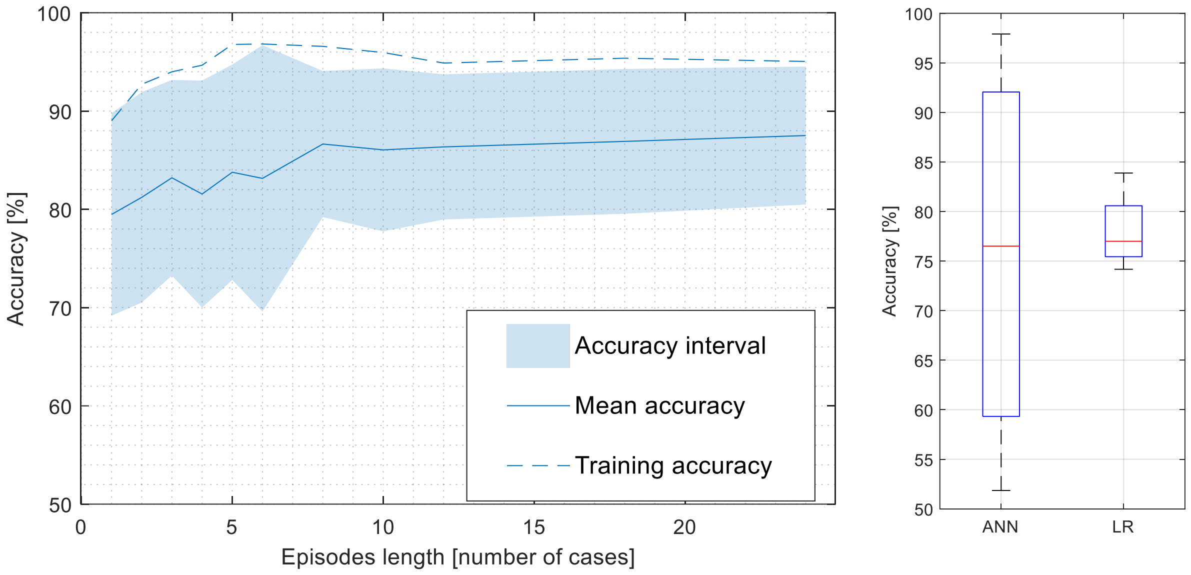


Figure 7.26 Accuracy of the temporal CBR solution (left); Accuracy for ANN and LR models (right)

Table 7.8 Performance of the TCBR approach: accuracy of the problem resolution for different datasets and comparison with ANN and LR models

Dataset	TCBR accuracy [%]											ANN	LR
	Episode length											accuracy	accuracy
	1	2	3	4	5	6	8	10	12	18	24	[%]	[%]
Training	89.00	92.72	93.99	94.66	96.77	**96.82**	96.58	95.95	94.88	95.37	95.04	**98.23**	78.32
Subset 0	91.69	93.56	95.25	94.62	**96.40**	95.90	95.79	95.87	95.69	96.08	95.24	**97.91**	74.17
Subset 1	78.40	80.81	84.19	80.96	84.47	86.93	**87.68**	87.08	86.18	84.55	83.98	66.80	**77.28**
Subset 2	66.63	67.50	70.93	66.63	69.62	63.98	77.74	75.63	77.63	78.50	**79.62**	51.85	**76.71**
Subset 3	81.18	83.01	82.45	84.00	84.58	85.77	85.36	85.60	85.90	**88.50**	81.18	**86.21**	83.88
Average	79,48	81,22	83,21	81,55	83,77	83,15	86,64	86,05	86,35	**86,91**	85,01	75,69	**78,01**

The values obtained with the TCBR method for each data set and for each episode length are shown on the left side of the Table 7.8. The best result for each dataset is highlighted in bold. On the right side, the same result is shown for the ANN and LR models. Again, the best result from both models is highlighted in bold, while the best result obtained from all the models is highlighted in blue. In the last row of the table the mean accuracy obtained from the 4 validation subsets (Subset 0 to Subset 3) is calculated. In the light of these results, the use of episodes instead of cases can be considered to improve the system performance. In comparison with other methods the performance obtained with the TCBR approach is similar or better for this case study.

Thus, the TCBR approach could be deployed in the real installation and used as an alternative to the CBR solution, providing better performance than the former. The task of implementing this method at production scale is easy due to the architecture proposed in Chapter 3 and the reasoning scheme in Chapter 4. It is only necessary to make a case-to-episode transformation when the process data is obtained, before the retrieval phase. Indeed, when the length of the episodes is one, the TCBR approach is like the CBR one. A more detailed discussion is done in Section 8.2.

7.4. Validation and Imputation module

In this section some results to validate the imputation module are presented. In the Subsection 7.4.1, the optimization of the CBR-based method is evaluated using synthetic data from a WWTP benchmark. In Subsection 7.4.2 results using real data from the case-study WWTPs are shown. In this case, the CBR-approach is compared with the ANN and LR models, as well as with different ensemble methods proposed in Subsection 6.4.2.

7.4.1 Results using synthetic data

The models that are usually considered for characterizing WWTP processes are the ones developed by the International Water Association (IWA), known as Activated Sludge Models (ASM). The imputation module is evaluated first on the BSM1 benchmark for performance assessment (Alex et al., 2008). To allow the testing and evaluation using the benchmark some input data files have been developed representing different weather conditions with realistic variations of the influent. The models of this benchmark have been implemented in the MATLAB/Simulink platform by the Technical University of Denmark (DTU) and Lund University (LU) (Jeppsson and Pons, 2004, Flores-Alsina et al., 2012).

Model ASM1 has been used to generate a valid data set to validate the imputation module. Model ASM1 include organic matter and nitrogen removal processes. Other models, such as ASM2 include also the biological phosphorus removal (Henze et al., 2000). The model used to simulate the plant does not implements any automatic control, —i.e., it is implemented in manual open loop. The influent data file contains the influent characterization for 609 days with 15 minutes sample time. From the 21 state variables available, only seven are considered (Table 7.9), assuming a simplification of the model, as proposed in (Nejjari et al., 2022).

Table 7.9 Considered variables from ASM1 model

Variable	Definition	Units
S_{NH}	NH4+NH3 nitrogen	mg/l
S_{NO}	Nitrate and nitrite nitrogen	mg/l
S_{O}	Oxygen	mg/l
X_{COD}	Chemical Oxygen Demand	mg/l
X_{BA}	Active autotrophic biomass	mg/l
X_{BH}	Active heterotrophic biomass	mg/l
Q_{IN}	Influent flow	m^3/d

From the generated data-set, one-year data is considered. 70% of the dataset is used to calibrate episodes length and feature weights for the TCBR approach, the remaining 30% for validation. The problem is defined as in (6-4) and solved using GA. The target variable is S_{NH}. In Figure 7.27 the original data is compared with the predicted data for one of the features, in this case, the S_{NH} concentration. In the top, the predicted value is calculated assuming unfaulty data and the parameters obtained from the optimisation solution. In the chart below, the predicted value is calculated assuming episodes of length 1 and the same weight for all features, i.e., non-optimal parameters. As it can be observed, the use of episodes instead of cases and an optimal set of feature weights allows to obtain a good prediction of the target feature.

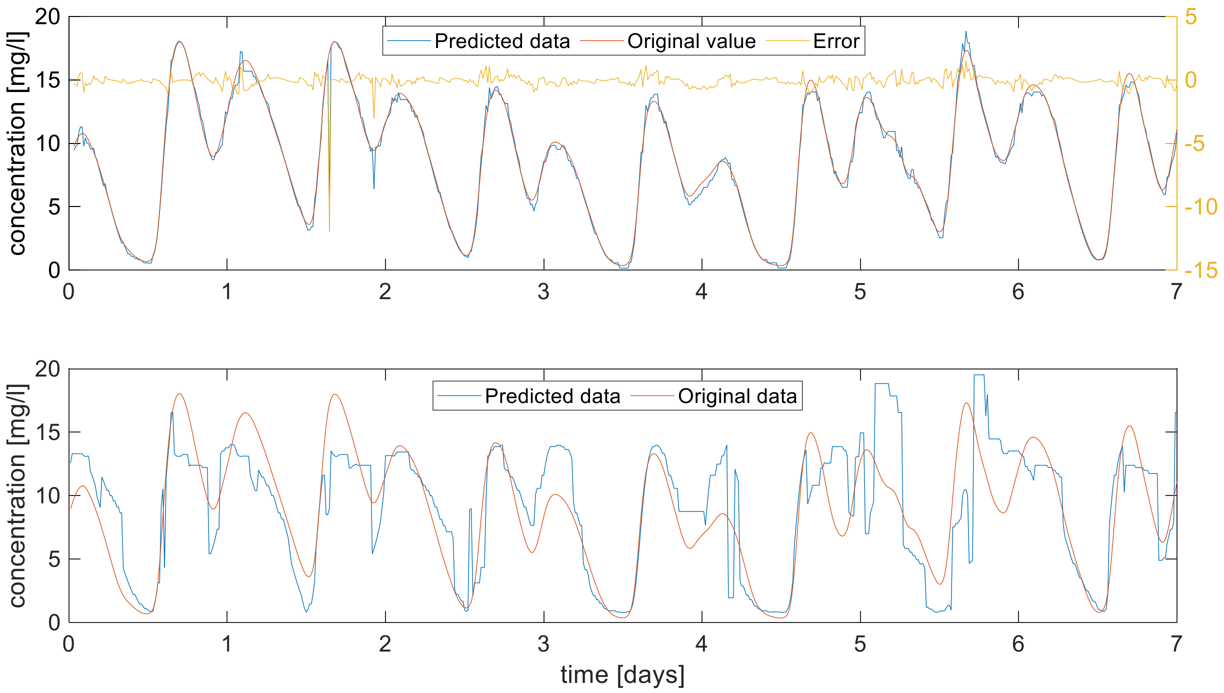


Figure 7.27 Calibration of episode length and feature weights for data imputation using TCBR and comparison with the non-calibrated module.

In Figure 7.28 predicted values are compared with the original ones, considering three different weeks from the validation dataset. As it can be observed, predictions one step ahead using unfaulty data are reliable enough to be considered for waste water treatment plant control.

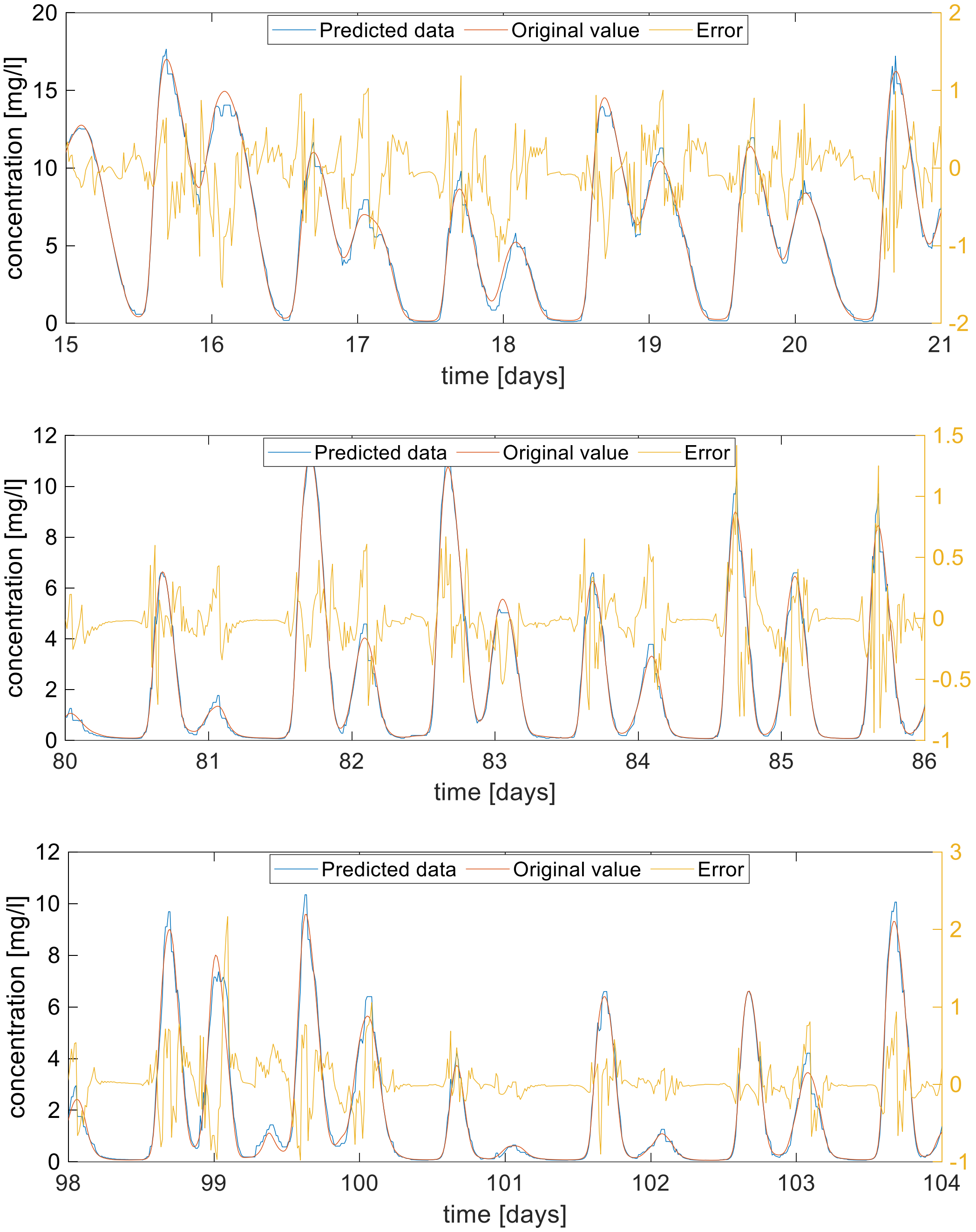
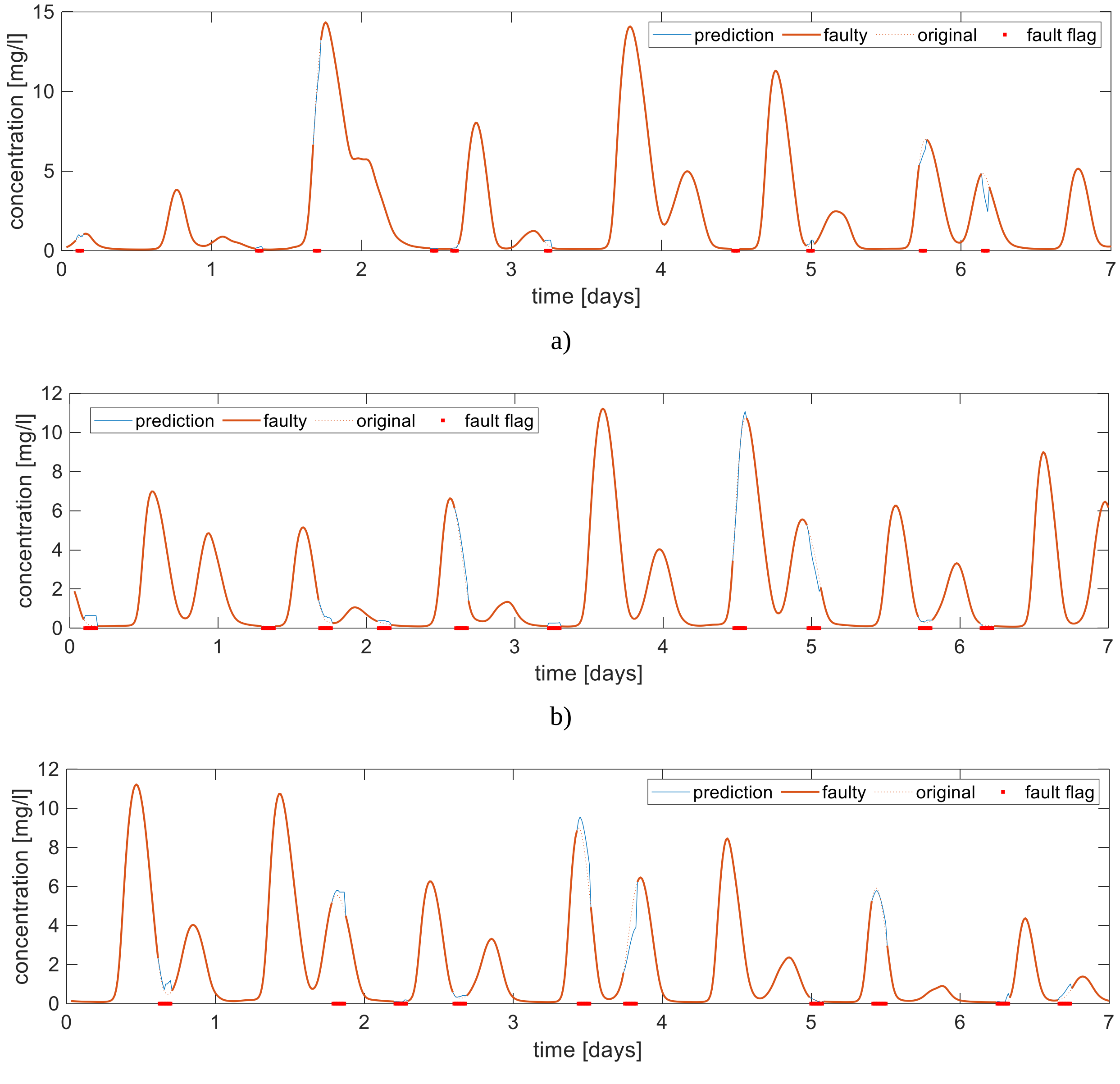


Figure 7.28 Test with unfaulty data considering three weeks from the validation dataset.

Then, to evaluate the performance of the obtained optimal parameters some synthetic faults are considered. Assuming that the sample time is 15 minutes and that this proposal is focused to solve the problem with short-time communication faults and/or outlier values, faults from four to eight samples are simulated, i.e., from one to two hours.



a)

b)

c)

Figure 7.29 Test with faulty data considering three weeks from the validation dataset: a) one hour missing value periods; b) and c) two hour missing value periods

Table 7.10 RMSE for different faults in Figure 7.29.

RMSE										
Fault	1	2	3	4	5	6	7	8	9	10
a) 1-hour faults	0.1349	0.1319	0.6116	0.0175	0.0334	0,3375	0,0190	0,2725	0,6471	1,4862
b) 2-hour faults	0.4798	0.2003	0.0284	0.1228	0.4270	0.1671	0.3709	0.6270	0.0735	0.0596
c) 2-hour faults	0.4168	0.4633	0.0412	0.0735	0.9873	1.2977	0.0260	0.1901	0.1391	0.2679

As it can be observed in Figure 7.29 and Table 7.10, the use of the TCBR method demonstrates promising results for missing data imputation. Most RMSE values are below 0.5, which it can be considered sufficient performance to be able to control the plant in a reliable way.

7.4.2 Results using real data from the case study

The ensemble model-based method proposed in Chapter 6 is tested with real data gathered from the WWTP described in Section 7.1. The IDSS is in operation in the real WWTP since January 2020. Data quality is paramount for models' calibration, as well as considering data from different situations or behaviours of the process, to obtain a reliable model able to impute missing values with a good performance. All models described in Section 6.4 are calibrated with a dataset complying with these requisites, and then, the obtained predictions are used for meta-predictors calibration. Then, all models are evaluated using a validation dataset: first, with unfaulty data and, then, considering different short and medium length faults ranging from one sample (5 minutes) to 12 samples (1 hour). In order to provide a convenient evaluation setup, faults are simulated —hence measured values are still available for performance testing. Simulated faults are based on typical faults occurring in the real facility —e.g., communication faults or invalid values during the sensor calibration process (a common sensor maintenance periodic procedure performed in the real facility)—, which make them representative of the incidences occurring in the operation of the real WWTP.

To evaluate the performance of the different models, the Root Mean Square Error (RMSE) is used. As it can be seen in Table 7.11, RMSE is calculated for different time periods, namely: the whole validation dataset ($RMSE_T$); the moving horizon window before the fault occurs ($RMSE_6$) and the fault time range ($RMSE_F$). Note that here faults are simulated by introducing missing values in the real data in order to calculate the error during the fault period. Figure 7.30 shows the performance of different models using an unfaulty dataset. The measured value is compared with the estimated one. Then, in Figure 7.31, the validation using faulty data is shown. Results shown in Figure 7.31 correspond to the fault #4 in Table 7.11. The fault length is 12 samples (1 hour) and starts where the $RMSE_6$ is depicted.

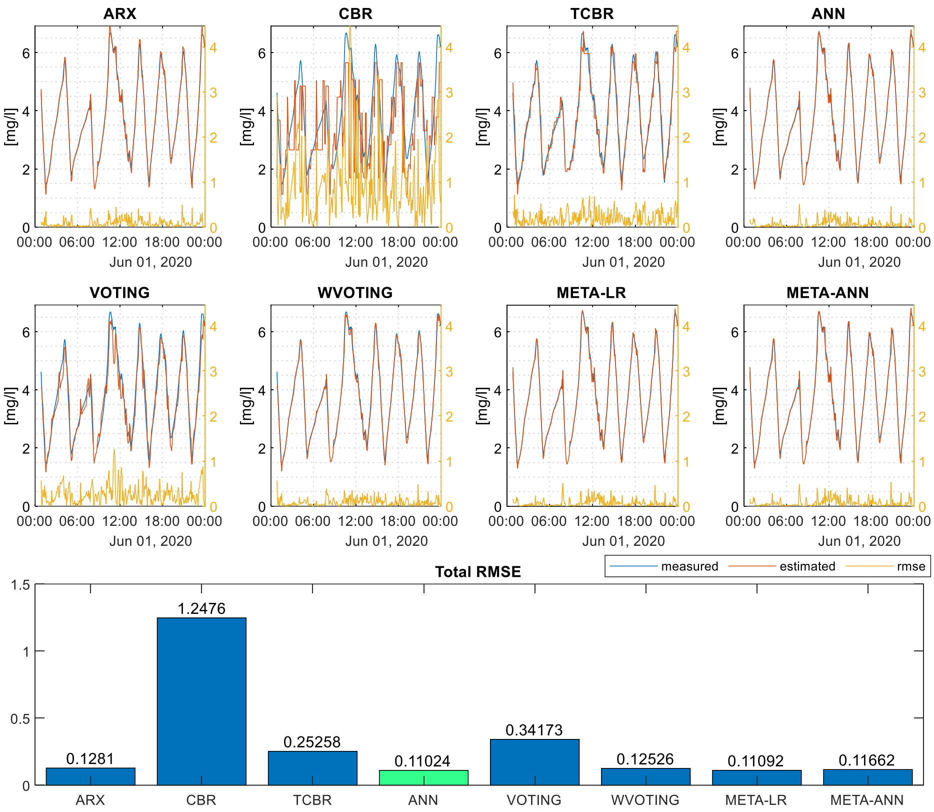


Figure 7.30. Comparison of measured and predicted values using different models with unfaulty data

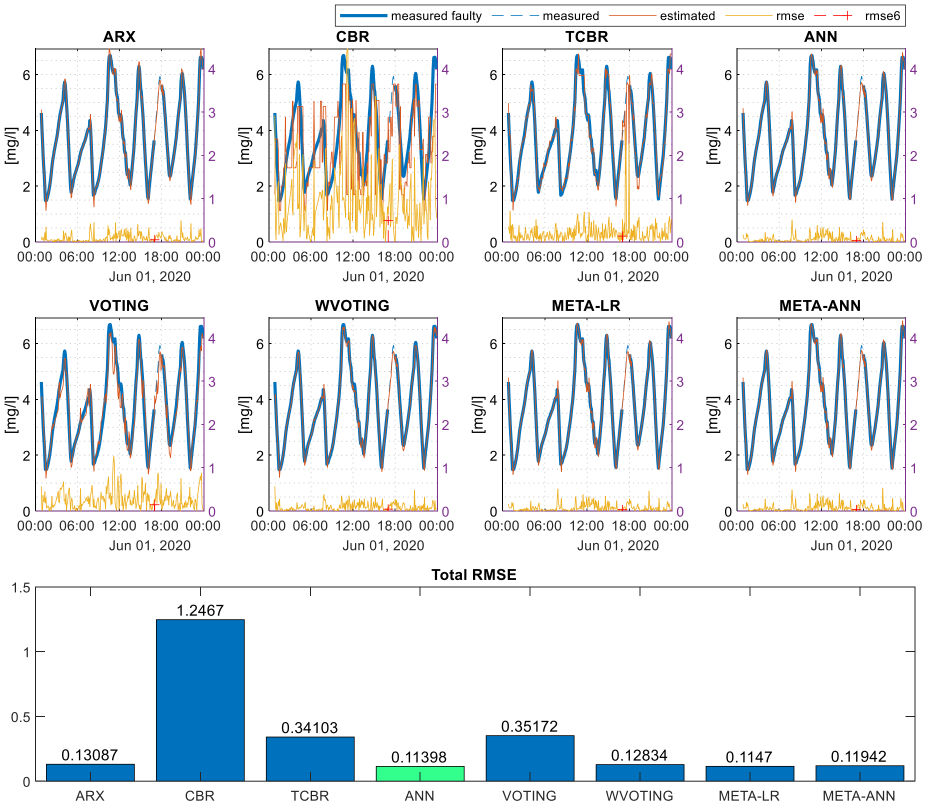


Figure 7.31. Evaluation of different models using faulty data (fault #4 in Table 7.11)

Table 7.11. Models' performance comparison for different faults in the ammonium sensor data

Id	Fault description	Length [samples]	Model Indicator	ARX	CBR	TCBR	ANN	Voting	Weighted voting	LR MP	ANN MP
#1	Unfaulty data	0	RMSE$_T$	0.1142	1.0942	0.1812	**0.0965**	0.2911	0.1065	**0.0967**	0.0997
#2	Short length, denitrification	3	RMSE$_T$	0.1301	1.2467	0.2501	**0.1123**	0.3717	0.1263	**0.1130**	0.1184
			RMSE$_G$	0.0425	0.8046	0.1802	**0.0124**	0.1531	0.0159	**0.0128**	0.0173
			RMSE$_F$	**0.0751**	0.4558	0.4057	0.1045	0.1772	**0.0764**	0.1018	0.1055
#3	Medium length from nitrification to denitrification	6	RMSE$_T$	0.1355	1.2467	0.2783	**0.1205**	0.3447	0.1332	**0.1211**	0.1250
			RMSE$_G$	0.0566	0.8823	0.1364	**0.0439**	0.2330	0.0511	**0.0429**	0.0458
			RMSE$_F$	**0.2587**	1.0414	0.9409	0.2820	0.4575	0.2856	0.2827	**0.2763**
#4	Medium length, from denitrification to nitrification	12	RMSE$_T$	0.1309	1.2467	0.3410	**0.1140**	0.3517	0.1283	**0.1147**	0.1194
			RMSE$_G$	0.0511	0.4977	0.1356	**0.0306**	0.1450	0.0437	**0.0315**	0.0319
			RMSE$_F$	0.1291	0.9651	1.1625	**0.1270**	0.4952	0.1502	**0.1282**	0.1333

7.5. Conclusions

In **Chapter 7** results are presented. First, in Section 7.1 the case study is described. A general context of the application domain is presented in Section 2.6, but here a more specific description of the particular case study is presented. Then, in the following sections results are shown. Section 7.2 describes how the proposed methodologies are integrated in a software IDSS tool. Following the concept in Chapter 3, all methods have been implemented in a general and scalable way to be easily deployed in any installation. It should be noted that the transfer process from research to real applications is critical to succeed in the implementation and user's adaptation. The development of a functional prototype of the IDSS proposed in this thesis has been achieved. Although there are still some lines of work, this prototype has been successfully installed and used in two real facilities with minor settings during the deployment stage. Distinct from using other ad-hoc solutions, here only the expert rules and the CB should be changed from one installation to another, if necessary. It is also easy to add or remove features from the CB. Nonetheless, the IDSS is directly connected to the PLC of the plant, so that it is essential that the PLC programming is aligned with the requirements of the IDSS.

In Section 7.3, experiments performed in real installations are detailed. The developed tool has been satisfactorily tested in two real facilities. The scalable design of all methods has provided a rapid set-up of the IDSS in both installations. The results for the first eight months of operation in the *Santa Maria de Palautordera* WWTP are described. To evaluate the performance of the proposed methods, some KPIs are proposed. These indexes are the ones usually used by plant managers and operators in order to supervise the process. The overall performance of the IDDS has been positively validated for a long period of operation, including many different situations, e.g. dry and rainy periods, low and high load influents. The baseline for the comparison is the performance before the IDSS. Obtained performance is quite similar to the one before the IDSS in terms of electrical consumption and nitrogen removal. It is the expected result as the combination of RBR and CBR methods tries to copy the human operator decisions.

The temporal CBR approach has been tested offline using real data, showing promising results in terms of accuracy in comparison with the static CBR approach. Regarding the episode's length, different lengths have been explored, observing that the accuracy increases until episode length eight, obtaining negligible improvement for higher length episodes. Hence, further work may be focused on the optimization of the episodes length to obtain the best performance, as proposed in Chapter 6 to improve the imputation performance. It would be interesting to explore how the optimization to minimize the imputation error and the optimization to find the best episode-length to provide an optimal outcome for the decision-making support can be related.

In Section 7.4 the imputation module is evaluated using synthetic and real data. A CBR model, considering the classic and temporal approaches is compared with classical models used for time-series prediction, such as ANN and LR. In addition, considering the heterogeneous nature of the measures involved in the process, different ensemble methods are tested aiming to improve the prediction performance in comparison with the one provided by individual models. First, the CBR-model is validated using synthetic data, and then, the whole methodology described in Chapter 6 is evaluated using real data from the WWTP. An improved performance has been observed using ensemble methods, achieving similar results as the ones obtained using ANN models. Also, the TCBR approach (episodes) outperforms the CBR solution (single cases).

8. Discussion

8.1. Discussion of the IDSS implementation

The developed methodologies proposed in this thesis have been integrated in an IDSS tool based on the concept proposed in Chapter 3. The described framework aims to enable the design of control and supervision systems for WWTPs, in an easily scalable way for the configuration of any plant. In this section, the work done in the implementation of each layer of the framework (Figure 3.2) is presented. The main functionalities of each layer have been tested in real installations described in Section 7.1:

- Data validation and model discovery methods in Layer 1.
- A graphical user interface in Layer 2, allowing the user to configure the reasoning process and supervising the plant with the help of some KPIs.
- A reasoning system for control set-points generation in Layer 3.

This first functional prototype has been developed in the MATLAB-Simulink environment, in combination with some external tools: an OPC Server to collect online data from the process, and a PostgreSQL database to store these data and relevant information for the operation of the tool. Although MATLAB-Simulink is not a common tool used in industry, especially in small and medium companies or local administrations, it has been very useful in the framework of this project, as it is a collaboration between the company and university. Considering that the work develop in this thesis is a strategic project for the company, in future stages it will be assessed whether MATLAB-Simulink is a suitable solution for the deployment of the final version of this tool or not. Many issues should be considered in order to find the best solution, such as cost, requirements from IT department, integration with other software, etc.

The visual configuration level in layer 2 has been left outside the scope of this thesis. The initial idea was to provide a block diagram environment to connect graphically data and models in layer 1 with the reasoning methods in layer 3, with the objective of designing the solution for a particular system. In any case, all implementations have been done in a modular and scalable way to add this functionality in the future.

The design and implementation of this tool has involved different users (WWTP managers and operators) and experts from CBT, so that the requirements and feedback from them have been considered, not only from the point of view of the IDSS performance, but also its usability. The definition of software usability is not standardized. For example, according to the standards for Ergonomics of human-system interaction by the International Organization for Standardizations (ISO9241-210:2019), usability is the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. At this stage, users and experts have been involved in the project, but any formal evaluation of the usability has been made. In the future, it would be interesting to consider state-of-the-art procedures for software evaluation tools.

To conclude, it is important to highlight the problems encountered during the implementation process. The implementation of solutions in real facilities is challenging because of many non-ideal situations that not occur when working, for example, in a simulation environment. The first problem encountered was the reliability of data from the process. Although large volumes of data are stored from real processes, many times data don't have the adequate quality. The second problem is the availability of data. While most processes in a WWTP are automated, sometimes there is a lack of sensors that does not allow to extract the full potential of the proposed methods. Finally, the last problem to tackle is the user's confidence in the proposed solutions and the resistance to change. In this sense, the use of explainable AI techniques has been positive valued by the users.

8.2. Discussion of the IDSS operation in real facilities

The KPIs designed to evaluate the performance of the tool provide useful information to the operators on the efficiency of the plant operation and the effluent quality, particularly in terms of nitrogen removal. The 24MA-AC KPI represented in Figure 7.22 shows that the ammonium concentration in the effluent (lab value) is predominantly below the allowed limit (only two samples out of 88 obtained from laboratory analitycal tests are over this limit). The online measure obtained from the ammonium sensor in the biological reactor (NH4) is used to control the process and gives a good aproximation of the concentration of this parameter in the effluent. It is assumed that a value within the limits in the biological reactor results in a value within the limits in the effluent.

In terms of total nitrogen (TNC KPI, Figure 7.24), most analytical results obtained in the laboratory are below the maximum allowed concentration. Only four samples out of 91 obtained from laboratory analitycal tests are out of bounds. These values over the allowed limit can be explained by the exceptional weather conditions in that period. From 19th to 25th of January 2020 a storm named *Gloria* was moving across Spain. The abundance of rainfall produced floods and the increase of influent flow in the plant. Rain water has a high concentration of DO, which supposes an increase of the nitrate's concentration and consequently, an increase of the total nitrogen.

The electrical consumption needed to achieve a good performance in nitrogen removal is compared to the consumption in the same period a year before the implementation of the tool presented in this work (Figure 7.23). The average values are quite similar, around 700 kWh/day, but it is necessary to consider some facts in the analysis of this result. The effluent quality is generally better than required by the current legislation, i.e. if the ammonium limit is 4 mg/l a lower concentration in the effluent is not required, largely due to some restrictions imposed by the operator. Hence, the operation in terms of electrical consumption may be improved by the relaxation of these restrictions, i.e. reducing the effect of the time-limits for the nitrification and denitrification cycles. Finally, in the period from May 2020 to August 2020, it can be observed an increase in the consumption as a consequence of an increase of the contamination in the influent, producing higher requirements for the aeration. One blower cannot provide enough oxygen to reach the set-point; hence, one backup blower is activated. Up until then the plant was operated using only one blower.

Considering these results and remarks, it should be noted that the nitrogen removal target is achieved. From the point of view of the electrical consumption it is difficult to compare the results obtained with previous historical data because of several changes in the process, e.g. the control is based on different sensors or the increase of oxygen demand. But despite all of that, electrical consumption is similar to the historical one and can be reduced addressing two points, namely: a) the adjustment of the nitrogen removal to the allowed limit; b) the reduction or removal of the restrictions that avoid the use of the solutions proposed by the reasoning cycle.

Before the installation of this IDSS, the redox measure sensor was used to control the biological process. Redox is an indirect measure to determine the end of nitrification and denitrification stages, whilst the ammonium and nitrate are direct measures. Thus, the control is expected to be more precise using the new ammonium and nitrate sensors, and therefore the electrical consumption could be reduced if some constraints imposed by the operator are relaxed.

The competence of the reasoning system is summarized in the Table 7.7. During the first three months most indexes cannot be determined. From January 2020 to the end of February 2020 only the RBR module is working. In March 2020 the CBR module is activated, but CBR_i and RBR_i indexes are not determined due to the detection and solution of several bugs during these first weeks. From April 2020 to August 2020 the application operation is considered stable and only minor bugs are detected and solved. Considering the values shown in Table 7.7, it can be observed a wide scope of the case base, i.e. most of the situations occuring are included in the CB, with a 99.97% of cases solved using the case-based reasoning module. It can be also noted that the E_i index is reduced from May 2020 due to increased user confidence on the application. This fact has allowed reducing the restrictions imposed by the open-loop fixed timers, which ideally should be removed completely in the future since they correspond to an open-loop operation of the facility. It has also been shown how the RBR module may be used to solve the first cases when historical records are scarce or unexistent to inicialize the CBR module with a CB. The CBR module has been partially activated after the validation of the RBR module with the retrieval, reuse and revision phases, in the period ranging from March to April. In April, after validation of the good performance of the initial case base, the retain phase is activated, together with the Decision module. Finally, the CSC index determines the percentage of correct solved cases by the whole reasoning system (i.e. RBR and CBR modules), taking into account the expert feedback, i.e. whether the reasoning system response is correct considering the expert criteria. For the period from April 2020 to August 2020 the percentage of correct solved cases is 91.74%, which is assumed to be a good performance from the practitioner point of view. Additionaly, in light of the water quality in the effluent and the high percentage of correct solved cases, the results may be considered enviromentally satisfactory.

Finally, a temporal CBR approach is compared with the static one. In Figure 7.26, it can be observed that there is an increase of the mean accuracy for the solution (calculated from all validation subsets) from around 80% for the classical CBR approach —i.e., for single episodes' length— to an accuracy around 87% when using 8 or more length for episodes. Calculating the accuracy interval using the mean and standard deviation of all validation subsets, it can be noted that the interval is narrowest as the length of episodes increases, remaining stable from length 8. Hence, in the light of these results, a good trade-off between accuracy, computing time and EB size —they increase with the length of episodes considered— is achieved for an episode length of 8 cases, with mean accuracy around 87% and a negligible difference when compared with the best accuracy attained with longer episode lengths. The EB size is increased by 50% when using length 12. Although in this particular case, for the required speed response and the sample time (5 minutes), any of the solutions could be implemented without compromising the solution computation in real operation conditions, computational issues should not be neglected when working with huge amounts of data in real systems. It is a good practice to make efficient use of computational resources and storage capacity. Regarding the comparison between different data subsets, it should be noted that the training accuracy is significantly higher than the one obtained with validation subsets. It can be explained because of some differences in the configuration of the plant between the training period and the validation period. During 2020 (training and validation Subset 0) only one biological reactor was operating. During 2022 (Subset 1, Subset 2 and Subset 3) both biological reactors were working. This fact may have affected to the process dynamics. When comparing the TCBR approach with an ANN model and a LR model (Figure 7.26), lower accuracy values are obtained. In addition, when using the ANN, the standard deviation is high, with validation accuracy values from almost 98% to 52%. Considering this and from the practical point of view, the explainability of a CBR-based model is an important advantage with regard to other a black-box models such as the ANN models, which does not easily provide any explanation about why a particular output is given. Precisely, the explainability of the method, as well as the scalability, are highly valued for the users supervising the process. Concerning the integration to a real facility, it should be noted that CBR approach make it possible to easily scale this solution to any plant configuration, including the addition of new available data.

To conclude, it is important to highlight that the use of CBR and RBR methods implemented in this work provides an explainable and reliable solution for the WWTP managers and operators. The explainability of the proposed solution enhances user's confidence in the daily use of AI techniques, as well as facilitate the understanding of the decisions made. In spite of this, the combination of explainable approaches such as the ones presented and other black-box methods would be very interesting to improve the whole system performance taking advantage of the strengths of these methods, such as the capacity of modelling complex processes or its efficiency

8.3. Discussion of the Validation and Imputation module

The proposed imputation methodology has been evaluated using first, synthetic data generated using WWTP models, and second, with offline real data from the WWTPs described in Section 7.1.

Synthetic data have been used to better illustrate the performance of the TCBR-based method (Subsection 7.4.1). When using real data, data quality or missing data can decrease the performance of the method, in the same way that it could happen with the other methods tested. To avoid such real problems and facilitating the optimization problem, synthetic data from BSM1 benchmark have been used. The evaluation of the TCBR method with synthetic data demonstrates promising results for missing data imputation.

Then, in Subsection 7.4.2, the methodology presented in Chapter 6 has been evaluated using real data from a WWTP and considering different realistic medium-range missing data windows based on real faults. In particular, the presented results are focused on the prediction of the ammonium sensor values, which is one of the most important measured variables for the biological process control of the WWTP.

The RMSE of the estimation with unfaulty data shows that the best model, i.e., the one with the lowest RMSE, is the ANN model. On the other hand, the performance of the CBR-based method is rather unsatisfactory when considering singular cases, but when using episodes, i.e., at least, two consecutive cases, the performance is similar to the one obtained with other methods. Furthermore, TCBR solutions have an advantage over other methods: they can increase their performance along time, as new relevant episodes can be learnt in the future. Those new episodes can help to an improved missing values imputation in the future. Concerning ensemble methods, similar results are obtained with *WV*, *LR-MP* and *ANN-MP* models, although the one showing the best performance is the *LR-MP*. It is also noteworthy the improvement between these methods and the *Voting* one.

Regarding results with faulty data, a similar prediction performance is obtained for all simulated faults, from 3 to 12 samples length. In general, the smallest RMSE is obtained using the *ANN*, regardless of whether the RMSE is evaluated for the whole period ($RMSE_T$) or during the fault ($RMSE_F$), while the best ensemble is the *LR-MP*. The higher performance degradation between the $RMSE_T$ and $RMSE_F$ can be observed for the 6 samples length fault.

To conclude, the most important points to highlight from obtained results are:

- The Temporal CBR-based approach (i.e. considering multiple consecutive cases) clearly outperforms the CBR-based approach (i.e., considering a single case).
- The feature weights calibration using GA allows to obtain a good prediction of ammonium sensor based on other present and past sensor values. The episode length has been also included as an optimization variable.
- With faulty data an acceptable imputation performance is obtained in most of the cases for short to medium length faults (i.e. during up to two hours).
- Although the T-CBR method does not outperforms other methods such as ARX or ANN models, it can provide a quite similar performance which is enough for process control purposes. In addition, T-CBR solutions have an advantage over other methods: they can increase their performance along time, as new relevant episodes can be learnt in the future. Those new episodes can help to an improved missing values imputation in the future.

9. Concluding Remarks

9.1. CONCLUSIONS

The present thesis focused on the development of an IDSS for assisting WWTPs operators in setting the main operational set-points of the treatment process. From the operational point of view, the proposed tool aims to consider real-time information from sensors in the process to guarantee the quality of the treated water at all times. But the design of this IDSS goes beyond the performance of the waste water treatment. The proposed architecture of the IDSS aims to solve a common challenge in the design and implementation of this type of tools for particular systems, which is the deployment of ad-hoc solutions for each installation.

Besides, the proposal has been envisioned from the perspective of a generic IDSS due to its ability to be adapted to other applications outside the domain of waste water treatment. This goal has been achieved using a design based on the interoperation of RBR and CBR techniques. The use of these tools and the proposed architecture have allowed overcoming the mentioned challenge for the implementation of IDSS in real facilities. Nevertheless, this has not been the only challenge to face. It is important to note that it is itself challenging working on real facilities and with real data, where many non-ideal situations can occur that do not occur when working in simulation. In this sense, two main issues have been encountered during the development of this thesis. First, data availability or data quality. Many times, the quantity of data is not a problem, but its quality and reliability. This problem is especially important when working with sensor data in real time. Another problem is the availability of the data, not due to a failure, but because that data is not being measured. The data quality problem has been addressed with a data validation and imputation proposal, which is also based on CBR. Second, the user confidence in the proposed solutions and the resistance to change. When a solution is being used for a long time and with good performance, it is not easy to convince the user that the new solution can provide, at least, the same functionalities. Here, the use of explainable methods such as RBR and CBR has not only facilitated the design of a scalable solution but also has enhanced user's confidence in the use of AI techniques for the day-by-day operation of full-scale WWTP.

The case study corresponds to the CBT environment, a public local water administration in charge of the management of 27 sanitation systems, composed by the sewer network from municipalities to waste water treatment plants. Its area of action covers an area of 1200 km^2, with a population of almost half a million, located to the north of Barcelona. In particular, the methodologies proposed in this thesis have been implemented in two full-scale facilities, *Santa Maria de Palautordera* and *Castellar del Vallès* WWTPs.
The proposed IDSS tool has allowed to generate control set-points for the biological process in the two aforementioned facilities. The evaluation has been addressed using the usual KPIs used by plant operators. On the other hand, the validation and imputation method developed has shown promising results, allowing its integration in the IDSS in the near future. Finally, the importance of the temporal component in dynamic domains has been proved. When considering episodes (TCBR approach) instead of cases (CBR approach) the performance of both set-points generation and imputation of wrong or missing values is improved.

9.2. FUTURE WORK

Given the presented results, some future work lines have been identified:

- The implementation of the block diagram environment to connect graphically data and models in Layer 1 with the reasoning tools in Layer 3. In addition, future versions of this tool will consider the use of open source tools. This work is a strategic project for CBT, so it is possible that in the future it can be implemented in other facilities. This is a quite ambitious point. Of course, to succeed in this purpose, it is necessary the collaboration of the practitioner as well as to find some resources to deploy the proposed IDSS to the 27 WWTP managed by CBT. This deployment would take many years, and it would be linked to a 5 years plan for the standardization of all SCADA systems.
- Regarding the temporal approach, the tuning process of the length of episodes could be optimised, as well as better explore the comparison between fixed and variable-length approaches. This optimization has been done to improve the performance of the imputation module, so it would be interesting to explore how it works from the point of view of the set-points generation accuracy. This task would take around 3 months.
- The comparison of the proposed imputation methods with other state-of-the-art proposals. In the same line, the proposed validation and imputation methods could be tested for other types of faults, e.g. incipient faults, as well as the evaluation of imputation methods with longer or multiple faults. Here, the integration of other AI black-box methods could be an interesting research line to be explored in order to improve the detection of invalid values, but also the imputation. All this work would take from 6 months to one year.

In the forthcoming future, the temporal reasoning approach and the validation and imputation module will be integrated in the IDSS to be tested online in the real facilities.

References

REFERENCES

Aamodt, A. and Plaza, E. Case-based reasoning: fundamental is-sues, methodological variations and system approaches. AI Communications 7(1):39-59, 1994.

Ahmed, S.A., Shadia R.T., Hala, A.T., 2002. Development and Verification of a Decision Support System for the Selection of Optimum Water Reuse Schemes. Desalination 152 (1-3), 339–352.

Alex, J., Benedetti, L., Copp, Jb., Gernaey, K., Jeppsson, U., Nopens, I., Pons, M-N., Rieger, L., Rosen, C., Steyer, J-P. (2008). Benchmark Simulation Model no. 1 (BSM1). Report by the IWA Taskgroup on Benchmarking of Control Strategies for WWTPs.

Allen, James (2013). Maintaining Knowledge about Temporal Intervals. Commun. ACM. 26. 510-521. 10.1016/B978-1-4832-1447-4.50033-X.

Alterman, R., Adaptive planning. Cognitive Science 12:393-422, 1988.

Arcos, J.L. and Plaza, E. Reflection in NOOS: an Object-centered representation language for knowledge modelling. In IJCAI Workshop on Reflection and Meta-level architectures and their application in AI (IJCAI'95), pages 1-10, Montréal, 1995.

Argent, R.M. An Overview of Model Integration for Environmental Applications-components, frameworks and semantics. Environmental Modelling & Software 19:219-234, 2004.

Ashley, K.D., Modelling legal argument: reasoning with cases and hypotheticals. The MIT Press, 1990.

Athanasiadis, I. and Mitkas, P. (2004). Supporting the decision-making process in environmental monitoring systems with knowledge discovery techniques. In: Proceedings of the Knowledge-based Services for the Public Services Symposium, Workshop III: Knowledge Discovery for Environmental Management. KDnet, pp. 1–12.

Aulinas, M., Nieves, J.C., Cortés, U., Poch, M. Supporting decision making in urban wastewater systems using a knowledge-based approach, Environmental Modelling & Software, Volume 26, Issue 5, 2011, Pages 562-572, ISSN 1364-8152, https://doi.org/10.1016/j.envsoft.2010.11.009.

Ba-Alawi, Nam, K., Heo, S., Woo, T., Aamer, H., & Yoo, C. (2023). Explainable multisensor fusion-based automatic reconciliation and imputation of faulty and missing data in membrane bioreactor plants for fouling alleviation and energy saving. *Chemical Engineering Journal (Lausanne, Switzerland : 1996)*, *452*, 139220–. https://doi.org/10.1016/j.cej.2022.139220

Bain, W. Case-based reasoning: a computer model of subjective assessment. Ph. D. Dissertation. Dept. of Computer Science. Yale University, 1986.

Bakan, Bénédicte; Bernet, Nicolas; Bouchez, Théodore; Boutrou, Rachel; Choubert, Jean-Marc; Dabert, Patrick; Duquennoi, Christian; Ferraro, Vincenza; García-Bernet, Diana; Gillot, Sylvie; Mery, Jacques; Rémond, Caroline; Steyer, Jean-Philippe; Trably, Eric and Tremier, Anne. (2022). Circular economy applied to organic residues and wastewater: Research challenges. *Waste and Biomass Valorization*, *13*(2), 1267-1276.

Béraud, B., Steyer, J. P., Lemoine, C., Latrille, E., Manic, G., & Printemps-Vacquier, C. (2007). Towards a global multi objective optimization of wastewater treatment plant based on modeling and genetic algorithms. Water Science and Technology, 56(9), 109-116.

Bernardelli, A., Marsili-Libelli, S., Manzini, A., Stancari, S., Tardini, G., Montanari, D., Venier, S. (2020). Real-time model predictive control of a wastewater treatment plant based on machine learning. Water Science and Technology, 81(11), 2391-2400. doi:10.2166/wst.2020.298

Berthuex, P.M., Lai, M., Darjatmoko, D., 1987. A Statistics-based information and expert system for plant control and improvement. In Proceeding of 5th National Conf. on Microcomputers in Civil Engineering, (W.E. Carrol, editor), Orlando, Florida, 146-150.

Branting, Luther, Hastings, John. (1994). An Empirical Evaluation of Model-Based Case Matching and Adaptation.

Brockwell, P.J., Davis, R.A. (1991). Time Series: Theory and Methods. Springer Series in Statistics. Springer, New York, NY.

Brown, D., Aldea, A., Harrison, R., Martin, C., Bayley, I. Temporal case-based reasoning for type 1 diabetes mellitus bolus insulin decision support, Artificial Intelligence in Medicine, Volume 85, 2018, Pages 28-42, ISSN 0933-3657, https://doi.org/10.1016/j.artmed.2017.09.007.

Buchanan B. G. and Duda R. O. Principles of Rule-Based Expert Systems. In Yovits, M.C. (ed.), Advances in Computers, Vol. 22, pp: 163-216. New York: Academic Press (1983).

Caiafa, C. F., Sun, Z., Tanaka, T., Marti-Puig, P., & Solé-Casals, J. (2021). Machine learning methods with noisy, incomplete or small datasets. *Applied Sciences*, 11(9), 4132.

Campbell, Stephen L., Chancelier, Jean-Philippe, Nikoukhah, Ramine (2010). Modeling and Simulation in cilab/Scicos with ScicosLab 4.4

Capodaglio, A.G., Jones, H.V., Novotny V., Feng, X., 1991. Sludge bulking analysis and forecasting: application of system identification and artificial neural computing technologies. Water Research 25(10), 1217-1224.

Carbonell, J. Derivational analogy: a theory of reconstructive problem solving and expertise acquisition. Machine Learning vol. 2, 1986.

Castillo, A., Cheali, P., Gómez, V., Comas, J., Poch, M., Sin, G., 2016. An integrated knowledge-based and optimization tool for the sustainable selection of wastewater treatment process concepts. Environmental Modelling & Software. 84, 177–192. https://doi.org/10.1016/j.envsoft.2016.06.019

Chapman, Stephen J. (2020). Matlab Programming for Enginners, 6th Edition. Cengage learnng.

Chen, S. H., Jakeman, A. J., & Norton, J. P. (2008). Artificial intelligence techniques: an introduction to their use for modelling environmental systems. *Mathematics and computers in simulation*, *78*(2-3), 379-400.

Cheng, C. H., Chan, C.-P., Sheu, Y.-J., 2019. A novel purity-based k nearest neighbors imputation method and its application in financial distress prediction. Eng. Appl. Artif. Intell. 81, 283–299.

Choi, G. W., Chong, K. Y., Kim, S. J. and Ryu, T. S. SWMI: new paradigm of water resources management for SDGs. Smart Water, vol. 1, no. 1, pp. 1–12, 2016.

Cognitive Systems. ReMind Developer's Reference Manual. Boston, 1992.

Corchado, J. M., & Lees, B. (2001). A hybrid case-based model for forecasting. *Applied Artificial Intelligence*, *15*(2), 105-127.

Corominas, L., Garrido-Baserba, M., Villez, K., Olsson, G., Cortés, U., Poch, M., 2018. Transforming data into knowledge for improved wastewater treatment operation: A critical review of techniques. Environmental Modelling & Software 106, 89–103. https://doi.org/10.1016/j.envsoft.2017.11.023

Cortés, U., Sànchez-Marrè, M., Sangüesa, R., Comas, J., Rodríguez-Roda, I., Poch, M., Riaño, D., 2001. Knowledge Management in Environmental Decision Support Systems. AI Commun. 14, 3–12.

Côte, M., Grandjean, B. P. A., Lessard, P., Thibault, J., 1995. Dynamic Modelling of the Activated Sludge Process: Improving Prediction Using Neural Networks. Water Res., 29 (4), 995-1004.

Cugueró-Escofet, M. À., García, D., Quevedo, J., Puig, V., Espin, S., & Roquet, J. (2016). A methodology and a software tool for sensor data validation/reconstruction: Application to the Catalonia regional water network. Control Engineering Practice, 49, 159–172.

Cugueró-Escofet, Miquel À., Joseba Quevedo, Vicenç Puig, and Diego García. Inconsistent Sensor Data Detection / Correction: Application to Environmental Systems." In Proceeding of: IEEE World Congress on Computational Intelligence (IEEE WCCI, 2014), Beijing, China, 84–90.

Cugueró-Escofet, M.À.; Puig, V. Advances in the Monitoring, Diagnosis and Optimisation of Water Systems. *Sensors* 2023, *23*, 3256. https://doi.org/10.3390/s23063256

Czoagala, E., Rawlik, T., 1989. Modelling of a Fuzzy Controller with application to the Control of Biological Processes. Fuzzy Sets and Systems 31, 13-22.

De Mantaras, R. L., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., ... & Watson, I. (2005). Retrieval, reuse, revision and retention in case-based reasoning. The Knowledge Engineering Review, 20(3), 215-240.

De Mulder, C., Flameling, T., Weijers, S., Amerlinck, Y., Nopens, I., 2018. An open software package for data reconciliation and gap filling in preparation of Water and Resource Recovery Facility Modeling. Environmental Modelling & Software 107, 186–198. https://doi.org/10.1016/j.envsoft.2018.05.015

Di Biccari, C. and Heigener, D. Semantic modeling of wastewater treatment plants towards international data format standards. In 30 Forum Bauinformatik (Weimar, Germany), 2018, no. September, pp. 183–190.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 1857 LNCS.

Digital Twin Consortium. 2023. The Definition of a Digital Twin. https://www.digitaltwinconsortium.org/initiatives/the-definition%20of-a-digital-twin/

Di Nardo, A., Boccelli, D. L., Herrera, M., Creaco, E., Cominola, A., Sitzenfrei, R., & Taormina, R. (2021). Smart Urban Water Networks: Solutions, Trends and Challenges. *Water* 2021, 13, 501.

DMG (2014). The Data Mining Group (http://www.dmg.org) leads the development of the Predictive Model Markup Language (PMML). Current version PMML 4.2. February 2014.

Dolin,R.H. and Alschuler,L. (2011). Approaching semantic interoperability in Health Level Seven. Journal of American Medical Informatics Association, Vol. 18:99-103.

Elasri,H. and Sekkaki, A. . Semantic Integration process of Business Components to Support Information System Designers. Int. Journal of Web & Semantic Technologies 4(1):51-65, 2013.

Eliades, D. G., Polycarpou, M. M. A Fault Diagnosis and Security Framework for Water Systems, in *IEEE Transactions on Control Systems Technology*, vol. 18, no. 6, pp. 1254-1265, Nov. 2010, doi: 10.1109/TCST.2009.2035515.

Elsawah, S., Filatova, T., Jakeman, A. J., Kettner, A. J., Zellner, M. L., Athanasiadis, I. N., Hamilton, S. H., Axtell, R. L., Brown, D. G., Gilligan, J. M., Janssen, M. A., Robinson, D. T., Rozenberg, J., Ullah, I. I. T., & Lade, S. J. (2020). Eight grand challenges in socio-environmental systems modeling. *Socio-Environmental Systems Modelling*, *2*, 16226-16226.

Erl, T. Service-Oriented Architecture. A Field Guide to Integrating XML and Web Services. Prentice-Hall, 2004.

Feng, L., Ouedraogo, A., Manghee, S. and Danilenko, A. A primer energy efficiency for municipal water and wastewater utilities. Washington DC, USA, 2012.

Fdez-Riverola, F., & Corchado, J. M. (2004). Fsfrt: Forecasting system for red tides. *Applied Intelligence*, *21*, 251-264.

Flanagan, M.J., 1980. On the Application of Approximate Resaoning to the Control of Activated Sludge Process. In Proceedings of Joint Automatic Control Conference, ASME, San Francisco, CA.

Flores-Alsina, Xavier & Gernaey, Krist & Jeppsson, Ulf. (2012). Benchmarking biological nutrient removal in wastewater treatment plants: Influence of mathematical model assumptions. Water science and technology: a journal of the International Association on Water Pollution Research. 65. 1496-505. 10.2166/wst.2012.039.

Flores, A., Tito, H., Silva, C., 2019. CBRm: Case Based Reasoning approach for imputation of medium gaps. Int. J. Adv. Comput. Sci. Appl. 10, 376–382.

Fox, J. and Das, J. (2000). Safe and sound. Artificial Intelligence in Hazardous Applications. AAAI Press/The MIT Press.

Gall, R., Patry G., 1989. Knowledge-based system for the diagnosis of an activated sludge plant. In Dynamic Modelling and Expert Systems in Wastewater Engineering. (G. Patry and D. Chapman editors), Chelsea, MI. Lewis Publishers, 1989.

Gao, X.W. and Gao, A., 2021. COVID-CBR: A Deep Learning Architecture Featuring Case-Based Reasoning for Classification of COVID-19 from Chest X-Ray Images, in: Proceedings - 20th IEEE International Conference on Machine Learning and Applications, ICMLA 2021. pp. 1319 – 1324. https://doi.org/10.1109/ICMLA52953.2021.00214

Gibert, K., Sànchez-Marrè, M., Codina, V. (2010). Choosing the Right Data Mining Technique: Classification of Methods and Intelligent Recommendation. 5th International Congress on Environmental Modelling and Software (iEMSs 2010). iEMSs' 2010 Proceedings, Vol. 3, pp. 1940-1947.

Gibert, K., Izquierdo, J., Sànchez-Marrè, M., Hamilton, S.H., Rodríguez-Roda, I., Holmes, G., 2018a. Which method to use? An assessment of data mining methods in Environmental Data Science. Environmental Modelling & Software 110, 3–27. https://doi.org/10.1016/j.envsoft.2018.09.021

Gibert, K., Horsburgh, J., Athanasiadis, I. and Holmes, G., 2018b. Environmental data science. *Environmental Modelling and Software* 106, 4–12.

Goel and B. Chandrasekaran, Case-based design: a task analysis. In Artificial Intelligence approaches to Engineering design, vol. 2: Innovative design (C. Tong and D. Sriram, editors). Academic Press, 1992.

Godo-Pla, L., Emiliano, P., González, S., Poch, M., Valero, F., Monclús, H. (2020). Implementation of an environmental decision support system for controlling the pre-oxidation step at a full-scale drinking water treatment plant. Water Science and Technology, 81(8), 1778–1785. https://doi.org/10.2166/wst.2020.142

Gottinger, H. W. and Weimann, P. (1992). Intelligent decision support systems. Decision Support Systems 8:317-332, 1992.

Gourbesville, P. Key Challenges for Smart Water. Procedia Eng., vol. 154, pp. 11–18, 2016.

Hamed, M.M., Khalafallah, M.G., Hassanien, E.A., 2004. Prediction of Wastewater Treatment Plant Performance Using Artificial Neural Networks. Environmental Modelling & Software 19, 919-928.

Hammond, K. Case-based planning: viewing planning as a memory task. Academic Press, 1989.

Han, H.-G., Zhang, H.-J., Liu, Z., Qiao, J.-F., 2020. Data-driven decision-making for wastewater treatment process. Control Engineering Practice, 96, art. no. 104305. https://doi.org/https://doi.org/10.1016/j.conengprac.2020.104305

Han, H., Sun, M., Li, F. "Online Aware Synapse Weighted Autoencoder for Recovering Random Missing Data in Wastewater Treatment Process," in *IEEE Transactions on Artificial Intelligence*, doi: 10.1109/TAI.2023.3266742.

Han, H., Li, M., Qiao, J., Yang, Q., Peng, Y. "Filter Transfer Learning Algorithm for Missing Data Imputation in Wastewater Treatment Process," in *IEEE Transactions on Knowledge and Data Engineering*, doi: 10.1109/TKDE.2023.3270118.

Health Level Seven (HL7). http://www.HL7.org

Hennessy, D.H. and Hinkle, D. Applying case-based reasoning to autoclave loading. IEEE Expert 7(5):21-26, 1992.

Henze, M., Gujer, W., Mino, T.,van Loosdrecht, M. (2000). Activated Sludge Models ASM1, ASM2, ASM2D, ASM3. 10.2166/9781780402369.

Hernandez-del-Olmo, F., Gaudioso, E. (2011). Reinforcement Learning Techniques for the Control of WasteWater Treatment Plants. In: Ferrández, J.M., Álvarez Sánchez, J.R., de la Paz, F., Toledo, F.J. (eds) New Challenges on Bioinspired Applications. IWINAC 2011. Lecture Notes in Computer Science, vol 6687. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21326-7_24

Herrera-Vega, J., Orihuela-Espina, F., Ibargüengoytia, P. H., García, U. A., Vila Rosado, D.-E, Morales, E. F., Sucar, L. E. A local multiscale probabilistic graphical model for data validation and reconstruction, and its application in industry, Engineering Applications of Artificial Intelligence, Volume 70, 2018, Pages 1-15, ISSN 0952-1976

Herrera, M., Meniconi, S., Alvisi, S., Izquierdo, J. (Eds.), 2018. Advanced Hydroinformatic Techniques for the Simulation and Analysis of Water Supply and Distribution Systems. MDPI publishers Pages: VIII-370.

Hinrichs, T. R. Problem solving in open worlds: a case study in design. Lawrence Erlbaum, 1992.

Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping Chen, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, & Hexagon-ML (2019). The UCR Time Series Classification Archive. URL https://www.cs.ucr.edu/~eamonn/time_series_data_2018/

Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: a Compilation of IEEE Standard Computer Glossaries. New York, 1990.

Jackson, P. (1999). Introduction to Expert Systems. 3rd edition. Boston, MA: Addison-Wesley.

Jære, Martha, Aamodt, Agnar, Skalle, P., (2002). Representing Temporal Knowledge for Case-Based Prediction. 2416. 174-188. 10.1007/3-540-46119-1_14.

Jakeman, A.J., Letcher, R.A., & Norton, J.P. (2006). Ten iterative steps in development and evaluation of environmental models. *Environmental Modelling and Software*, 21, 602-614. https://doi.org/10.1016/j.envsoft.2006.01.004.

Jeppsson, U.; Pons, M.N. The cost benchmark simulation model current state and future perspective. Control Eng. Pract. 2004, 12, 299–304.

Jiang, Z., Jiang, Y., Wang, Y., Zhang, H., Cao, H., Tian, G., 2019. A hybrid approach of rough set and case-based reasoning to remanufacturing process planning. J. Intell. Manuf. 30, 19 – 32. https://doi.org/10.1007/s10845-016-1231-0

Johnson, G.W. and Jennings, R., 2006. LabVIEW graphical programming. McGraw-Hill, New York

Johnston, M. W., Hanna, J. R. P., Millar, R. J. (2004). Advances in dataflow programming languages. ACM Computer. Surveys 36. 1-34. DOI: 10.1145/1013208.1013209.

Karr, C.L., 1991. Genetic Algorithms for Fuzzy Controlers. AI Expert 6(2), 26-33.

Kass, A.M., and Leake, D.B. Case-based reasoning applied to constructing explanations. Proc. of Workshop on case-based reasoning (DARPA). Clearwater, Florida. 1988.

Kokkinos, K., Karayannis, V., Nathanail, E., Moustakas, K., 2021. A comparative analysis of Statistical and Computational Intelligence methodologies for the prediction of traffic-induced fine particulate matter and NO2. J. Clean. Prod. 328.

Kolodner, J.L. and Simpson, R.L. The MEDIATOR: analysis of an early case-based problem solver. Cognitive Science 13(4):507-549, 1989.

Kolodner, J.L. Memory for experience. In The psychology of learning and motivation vol. 19 (G. Bower editor). Academic Press, 1985.

Kolodner, J.L., Case-Based Reasoning. Morgan Kaufmann, 1993.

Komatsoulis G.A., Warzel, D.B., Hartel, F.W., Shanbhag, K., Chilukuri, Ram, Fragoso G., de Coronado S., Reeves, D.M., Hadfield, J.B., Ludet, C. and Covitz, P.A. caCORE version 3: Implementation of a Model Driven, Service-Oriented Architecture for Semantic Interoperability. Journal of Biomedical Informatics 41(1):1-29, 2008

Kosko, B., 1992. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice Hall, USA.

Koton, P. Using experience in learning and problem solving. Ph. D. dissertation. Dept. of Computer Science. MIT, 1989.

Kzaz, L., Elasri, H., and Sekkaki, A. (2010). A Model for Semantic Integration of Business Components. Int. Journal of Computer Science & Information Technology 2(1):1-12.

Laxmi, P. and G. Laxmi-Deepthi, G. Smart Water Management Process architecture with IoT Based Reference. Int. J. Comput. Sci. Mob. Comput., vol. 6, no. 6, pp. 271–276, 2017.

Leake, D.B., Kinley A., and Wilson, D. Case-based similarity assesment: estimating adaptability from experience. Proc. of American Association of Artificial Intelligence (AAAI-97), pp. 674-679, 1997.

Lupiani, Eduardo & Juarez, Jose & Palma, Jose & Marín, Roque. (2017). Monitoring Elderly People at Home with Temporal Case-Based Reasoning. Knowledge-Based Systems. 134. 10.1016/j.knosys.2017.07.025.

Ma J., Knight B. (2003) A Framework for Historical Case-Based Reasoning. In: Ashley K.D., Bridge D.G. (eds) Case-Based Reasoning Research and Development. ICCBR 2003. Lecture Notes in Computer Science, vol 2689. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45006-8_21

Mackay, D. S. (1999). Semantic Integration of Environmental Models for Application to Global Information Systems and Decision-Making. ACM SIGMOD Record 28(1):13-19, March 1999.

Maeda, K., 1989. A knowledge-based system for the wastewater treatment plant. Future Generation Computer Systems 5, 29-32.

Manguinhas, H. (2010). Achieving Semantic Interoperability using Model descriptions. Bulletin of IEEE Technical Committee on Digital Libraries, Vol. 6, No. 2, Fall 2010.

Mannina, G., Rebouças, T., Cosenza, A., Sànchez-Marrè, M. and Gibert, K. (2019). Decision support systems (DSS) for wastewater treatment plants – A review of the state of the art. Bioresource Technology. Vol. 290. 121814. 10.1016/j.biortech.2019.121814.

Martí Navarro, M., De Paz, J. F., Julián, V., Rodríguez, S., Bajo, J., Corchado, J. M. Temporal bounded reasoning in a dynamic case based planning agent for industrial environments, Expert Systems with Applications, Volume 39, Issue 9, 2012, Pages 7887-7894, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2012.01.119.

Martín, Francisco & Plaza, Enric. (2004). Ceaseless Case-Based Reasoning. 3155. 287-301. 10.1007/978-3-540-28631-8_22.

Martí-Sarri, A., Serra-Serra, M., & Marti-Puig, P. (2019). Effect of the Data Tensorization on the Recovery of Bursts of Missing Values. An Application in Water Networks. In *Artificial Intelligence Research and Development* (pp. 245-255). IOS Press.

McIntosh, B.S., J.C. Ascough II , M. Twery , J. Chewe, A. Elmahdi , D. Haase , J.J. Harou , D. Hepting , S. Cuddy , A.J. Jakeman, S. Chen, A. Kassahun, S. Lautenbach, K. Matthews, W. Merritt , N.W.T. Quinnm, I. Rodriguez-Roda, S. Sieber , M. Stavenga, A. Sulis , J. Ticehurst, M. Volk, M. Wrobel, H. van Delden, S. El-Sawah, A. Rizzoli and A. Voinov (2011). Environmental decision support systems

(EDSS) development–Challenges and best practices. *Environmental Modelling & Software*, *26*(12), 1389-1402.

Meléndez, J.; de la Rosa, J.Ll.; Macaya, D.A.; Colomer Llinàs, J. 3r Congrés Català d'Intel.ligècia Artificial (CCIA). Vilanova i la Geltrú (ESP) 2000.; Case based approach for generation of recipes in batch process control.

Michel Jaczynski. 1997. A framework for the management of past experiences with time-extended situations. In Proceedings of the sixth international conference on Information and knowledge management. Association for Computing Machinery, New York, NY, USA, 32–39. DOI: https://doi.org/10.1145/266714.266851

Montani, Stefania & Portinale, Luigi. (2005). Case Based Representation and Retrieval with Time Dependent Features. 3620. 353-367. 10.1007/11536406_28.

Morrison, J.P., 2010. Flow-Based Programming: A new approach to application development. CreateSpace, 2010.

Mustafa, H. M., Mustapha, A., Hayder G., Salisu, A. "Applications of IoT and Artificial Intelligence in Water Quality Monitoring and Prediction: A Review," *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, 2021, pp. 968-975, doi: 10.1109/ICICT50816.2021.9358675.

Nadiri, A.A., Shokri, S., Tsai, F.T.C., Moghaddam, A. A., 2018. Prediction of Effluent Quality Parameters of a Wastewater Treatment Plant Using a Supervised Committee Fuzzy Logic Model. Journal of Cleaner Production 180, 539–549.

Nagar, S. (2017). Introduction to Scilab for Engineers and Scientists. https://www.scilab.org/

Nasiri, S. and Khosravani, M.R., 2019. Faults and failures prediction in injection molding process. Int. J. Adv. Manuf. Technol. 103, 2469–2484. https://doi.org/10.1007/s00170-019-03699-x

Nawaz A., et al., "Intelligent human-machine interface: An agile operation and decision support for ANAMMOX SBR system at a pilot-scale wastewater treatment plant", *IEEE Trans. Ind. Inform.*, vol. 18, no. 9, pp. 6224-6232, Sep. 2022.

Nejjari, F.; Khoury, B.; Puig, V.; Quevedo, J.; Pascual, J.; de Campos, S. Economic Linear Parameter Varying Model Predictive Control of the Aeration System of a Wastewater Treatment Plant. Sensors 2022, 22, 6008. https://doi.org/10.3390/s22166008

Ngouna, R.H., Ratolojanahary, R., Medjaher, K., Dauriac, F., Sebilo, M., Junca-Bourié, J., 2020. A data-driven method for detecting and diagnosing causes of water quality contamination in a dataset with a high rate of missing values. Eng. Appl. Artif. Intell. 95, 103822.

Núñez, H., Sànchez-Marrè, M., Cortés, U., Comas, J., Martínez, M., Rodríguez-Roda, I. and Poch, M. (2004). A comparative study on the use of similarity measures in case-based reasoning to improve the classification of environmental system situations. Environmental Modelling & Software 19, 809–819. https://doi.org/10.1016/j.envsoft.2003.03.003.

Oehmcke, S., Zielinski, O., Kramer, O., 2016. kNN ensembles with penalized DTW for multivariate time series imputation, in: Proceedings of the International Joint Conference on Neural Networks. pp. 2774–2781.

Olsson, G., Nielsen, M., Yuan, Z., Lynggaard-Jensen, A., & Steyer, J. P. (2005). *Instrumentation, Control and Automation in Wastewater Systems.* IWA Publishing.

Oprea, M. (2018). A knowledge modelling framework for intelligent environmental decision support systems and its application to some environmental problems. *Environmental Modelling & Software*, *110*, 72-94.

Oprea, M., & Dunea, D. (2010). SBC-MEDIU: a multi-expert system for environmental diagnosis. *Environmental Engineering and Management Journal*, *9*(2), 205-213.

Osborne, H. R. and Bridge, D. G. A case base similarity framework. Proc. of 4th European Workshop on Case-Based Reasoning (EWCBR'98), pp. 309-323, 1998.

Ouksel, A.M. and Sheth, A. (1999). Semantic Interoperability in Global Information Systems: a Brief Introduction to the Research Area and the Special Section. ACM SIGMOD Record 28(1):5-12, March 1999.

Oulebsir, R., Lefkir, A., Safri, A. and Bermad, A. Optimization of the energy consumption in activated sludge process using deep learning selective modeling, Biomass and Bioenergy, vol. 132, no. October 2019, p. 105420, 2020.

Pascual-Pañach, J., Cugueró-Escofet, M.A., Aguiló-Martos, P., Sànchez-Marrè, M. (2018a). An Interoperable Workflow-Based Framework for the Automation of Building Intelligent Process Control Systems. 9th International Congress on Environmental Modelling and Software, Fort Collins, Colorado, USA.

Pascual-Pañach, J., Cugueró-Escofet, M.A., Aguiló-Martos, P., Sànchez-Marrè, M. (2018b). Herramienta basada en minería de datos para automatización del diseño de sistemas inteligentes en EDAR. XXXV Congreso AEAS 2019, Valencia, Spain.

Pascual-Pañach, J., Cugueró-Escofet, M.A., Sànchez-Marrè, M., Aguiló-Martos Martos, P. (2019b). Data mining based tool for the automation of the design of intelligent process control systems in waste water treatment plants. IWA Spain National Young Water Professionals Conference, Madrid, Spain.

Pascual-Pañach, J., Cugueró-Escofet, M.A., Sànchez-Marrè, M., Aguiló-Martos, P. (2019a). Application of CBR for intelligent process control of a WWTP. IOS Press, Frontiers in artificial intelligence and applications, 2019, 319, 160 - 169, 0922-6389

Pascual-Pañach, J., Sànchez-Marrè, M., Cugueró-Escofet, M.A., (2022a). Ensemble model-based method for time series sensors' data validation and imputation applied to a real Waste Water Treatment Plant. 11th International Congress on Environmental Modelling and Software, Brussels, Belgium.

Pascual-Pañach, J., Sànchez-Marrè, M., Cugueró-Escofet, M.A., (2022b). Optimizing Online Time-Series Data Imputation Through Case-Based Reasoning. IOS Press, Frontiers in artificial intelligence and applications, 2022, 356, 87 – 90, 10.3233/FAIA220320

Pascual-Pañach, Josep, Cugueró-Escofet, Miquel Àngel, Sànchez-Marrè, Miquel, Interoperating data-driven and model-driven techniques for the automated development of intelligent environmental decision support systems, Environmental Modelling & Software, Volume 140, 2021, 105021, ISSN 1364-8152, https://doi.org/10.1016/j.envsoft.2021.105021.

Pascual-Pañach, Josep, Sànchez-Marrè, Miquel, Cugueró-Escofet, Miquel Àngel, (2022c). Temporal Case-Based Reasoning for Intelligent Environmental Decision Support Systems. Submitted to Engineering Applications of Artificial Intelligence in January 2022.

Pérez-Pons, M. E., Parra-Dominguez, J., Hernández, G., Bichindaritz, I., & Corchado, J. M. (2023). OCI-CBR: A hybrid model for decision support in preference-aware investment scenarios. *Expert Systems with Applications*, *211*, 118568.

Phillips-Wren, G., Mora, M., Forgionne, G. A. and Gupta, J. N. D. (2009). An integrative evaluation framework for intelligent decision support systems. European Journal of Operational Research 195:642–652, 2009.

Poch, M., Comas, J., Cortés, U., Sànchez-Marrè, M., Rodríguez-Roda, I. Crossing the death valley to transfer environmental decision support systems to the water market. "Global challenges", 10 Abril 2017, vol. 1, núm. 3, p. 1700009-1-1700009-10.

Poch, M., Comas, J., Rodríguez-Roda, I., Sànchez-Marrè, M., Cortés, U. (2004). Designing and building real environmental decision support systems. Environmental Modelling & Software 19: 857-873. 10.1016/j.envsoft.2003.03.007.

Qi, X., Guo, H., Wang, W., 2021. A reliable KNN filling approach for incomplete interval-valued data. Eng. Appl. Artif. Intell. 100, 104175.

Quesada, D., Valverde, G., Larrañaga, P., Bielza, C., 2021. Long-term forecasting of multivariate time series in industrial furnaces with dynamic Gaussian Bayesian networks. Eng. Appl. Artif. Intell. 103, 104301.

Ráduly, B., Gernaey, K. V, Capodaglio, A.G., Mikkelsen, P.S., Henze, M., 2007. Artificial neural networks for rapid WWTP performance evaluation: Methodology and case study. Environmental Modelling & Software 22, 1208–1216. https://doi.org/10.1016/j.envsoft.2006.07.003

Ram, Ashwin, Santamaría, Juan. (1997). Continuous Case-Based Reasoning. Artificial Intelligenge 90. 25-77. 10.1016/S0004-3702(96)00037-9.

Redmond, M.A. Learning by observing and understanding expert problem solving. Georgia Institute of Technology. College of Computing. Technical report GIT-CC-92/43, 1992.

Richter, M. and Weber, R. (2013). Case-based reasoning: a textbook. Springer

Riesbeck, C.K. and Schank, R.C.. Inside Case-Based Reasoning. Lawrence Erlbaum Associates Publishers, 1989.

Rizzoli, A.E. and Young, W.Y. (1997). Delivering environmental decision support systems: software tools and techniques. *Environmental Modelling and Software*, 12 (2-3), 237–249.

Rizzoli, A. E., Davis, J.R. and Abel, D.J. (1998). Model and Data Integration and re-use in Environmental Decision Support Systems. Decision Support Systems 24:127-144, 1998.

Robles, T., Alcarria, R., Martín, D., Navarro, M., Calero, R., Iglesias, S. López, M. 2015. An iot based reference architecture for smart water management processes. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications. 6. 4-23.

Ruano, M. V, Ribes, J., Sin, G., Seco, A., Ferrer, J., 2010. A systematic approach for fine-tuning of fuzzy controllers applied to WWTPs. Environmental Modelling & Software 25, 670–676. https://doi.org/10.1016/j.envsoft.2009.05.008

Ruiz Gutierrez, J. M., 2017. https://myopenlab.org/documentos/

Saad, M., Nassar, L., Karray, F., Gaudet, V., 2020. Tackling Imputation Across Time Series Models Using Deep Learning and Ensemble Learning, in: Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics. pp. 3084–3090.

Sacerdoti, E.D. A structure for plans and behavior. North-Holland, 1977.

Salantino, M., De Maio, M., Aliverti, E. (2016). Mastering JBoss Drools 6. Packt.

Sànchez-Marrè, M. (2022). Intelligent Decision Support Systems. Springer Nature, Switzerland AG, March 2022. ISBN: 978-3-030-87789-7. DOI: https://doi.org/10.1007/978-3-030-87790-3

Sànchez-Marrè, M., 2014. Interoperable Intelligent Environmental Decision Support Systems: a Framework Proposal. 7th International Congress on Environmental Modelling & Software (iEMSs 2014). iEMSs 2014 Proceedings, 1, 501-508. Ames, D.P., Quinn, N.W.T., Rizzoli, A.E. (Eds.), 201

Sànchez-Marrè, M., Cortés, U., R.-Roda, I., Poch, M. L'Eixample distance: a new similarity measure for case retrieval. Procc. of 1st Catalan Conference on Artificial Intelligence (CCIA'98), pp. 246-253.Tarragona, Catalonia, EU, 1998.

Sànchez-Marrè, M., Cortés, U., Rodríguez-Roda, I., Poch, M., Lafuente, F., 2002. Learning and Adaptation in Wastewater Treatment Plants Through Case–Based Reasoning. Comput. Civ. Infrastruct. Eng. 12, 251–266. https://doi.org/10.1111/0885-9507.00061

Sànchez-Marrè, M., Cortés, U., R-Roda, I., Poch, M., Lafuente, J., 1997. Learning and Adaptation in WWTP through Case-Based Reasoning. Special issue on Machine Learning. Microcomputers in Civil Engineering/Computer-Aided Civil and Infrastructure Engineering 12(4), 251-266.

Sànchez-Marrè, M., Gibert, K., Sojda R., Steyer J.P., Struss, P. and Rodríguez-Roda, I. (2006) Uncertainty Management, Spatial and Temporal Reasoning and Validation of Intelligent Environmental Decision Support Systems. 3rd International Congress on Environmental Modelling and Software (iEMSs'2006). iEMSs' 2006 Proceedings, pp. 1352-1377.

Sànchez-Marrè, M., Martinez, M., Rodríguez-Roda I., Alemany, J., Cortés, C., 2004. Using CBR to improve intelligent supervision and management of wastewater treatment plants: the atl_EDAR system. 7th European Conference on Case-Based Reasoning (ECCBR'2004), Proc. of Industrial day, 7th European Conference on Case-Based Reasoning (Eds. Francisco Martin and Mehmet Göker), 79-91

Sànchez-Marrè, Miquel, Cortés, Ulises, Martínez, Montse, Comas, Joaquim, Rodríguez-Roda, Ignasi. (2005). An Approach for Temporal Case-Based Reasoning: Episode-Based Reasoning. Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science). 3620. 465-476. 10.1007/11536406_36.

Santín, I., Barbu, M., Pedret, C., Vilanova, R., 2018. Fuzzy logic for plant-wide control of biological wastewater treatment process including greenhouse gas emissions. ISA Trans. 77. https://doi.org/10.1016/j.isatra.2018.04.006

Santos, G., Pinto, T., Vale, Z., & Corchado, J. M. (2021). Semantic Interoperability for Multiagent Simulation and Decision Support in Power Systems. Communications in Computer and Information Science, 1472 CCIS, 215–226. https://doi.org/10.1007/978-3-030-85710-3_18

Schank, R.. Dynamic memory: a theory of learning in computers and people. Cambridge University Press, 1982.

Schmidt, Rainer & Gierl, Lothar. (2002). Applying temporal abstraction and case-based reasoning to predict approaching influenza waves. Studies in health technology and informatics. 90. 420-4. 10.3233/978-1-60750-934-9-420.

Serrà, J., Arcos, J.L., 2012. A Competitive Measure to Assess the Similarity between Two Time Series. Case-Based Reasoning Research and Development, in: Agudo, B.D., Watson, I. (Eds.), Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 414–427. https://doi.org/10.1007/978-3-642-32986-9_31

Serra, P., Sànchez-Marrè, M., Lafuente, J., Cortés, U. and Poch, M., 1994. DEPUR: a knowledge based tool for wastewater treatment plants. Engineering Applications of Artificial Intelligence 7(1), 23-30.

Shinn, H.S., Abstractional analogy: a model of analogical reasoning. Proc. of Workshop on case-based reasoning (DARPA). Clearwater, Florida. 1988.

Sottara, D., Bragaglia, S., Mello, P., Pulcini, D., Luccarini, L and Giunchi, D. (2012). Ontologies, Rules, Workflow and Predictive Models: Knowledge Assets for an EDSS. 6th International Congress on Environmental Modelling and Software (iEMSs'2012). iEMSs' 2012 Proceedings, pp. 204-211.

Steels, L. Components of expertise. AI Magazine 11(2):28-49, 1990.

Sussman, G.J. A computer model of skill acquisition. American Elsevier, 1975.

Sycara, K. Finding creative solutions in adversarial impasses. Proc. of 9th Annual Conference of the Cognitive Science Society. Lawrence Erlbaum, 1987.

Syu, M.-J. and Chen. B.-C., 1998. Back-propagation Neural Network Adaptive Control of a Continuous Wastewater Treatment Process. Industrial & Engineering Chemistry Research, 37(9), 3625-36230.

Torregrossa, D., Hernández-Sancho, F., Hansen, J., Cornelissen, A., Popov, T., Schutz, G., 2017. Energy Saving in Wastewater Treatment Plants: A Plant-Generic Cooperative Decision Support System. Journal of Cleaner Production 167, 601–609.

Tzafestas, S. and Ligeza. A., 1989. A Framework for Knowledge Based Control. Intelligent and Robotic Systems 1(4), 407-426.

Veloso, M.M. and Carbonell. J.G. Derivational Analogy in PRODIGY: automating case acquisition, storage and utilization. Machine Learning 10(3):249-278, 1993.

Vetere, G. and Lenzerini, M. (2005). Models for Semantic Interoperability in Service-oriented architectures. IBM Systems Journal 44(4):887-903, 2005.

Walczak, S., Cerpa, N., 2003. Artificial Neural Networks, Encyclopedia of Physical Science and Technology (Third Edition). Academic Press, New York, pp. 631–645.

Wang, X.Z., Chen, B.H., Yang, S.H., McGreavy, C., Lu, M.L. , 1997. Fuzzy Rule Generation from Data for Process Operational Decision Support. Computers Chem. Engng. 21, S661-S666

Wang, Y., Shen, Y., Liu, J., Zhou, X., Wu, X., Chen, B. "RFE-LSTM-Based Effluent Quality Prediction Method for Wastewater Treatment Plant," *2022 IEEE 31st International Symposium on Industrial Electronics (ISIE)*, Anchorage, AK, USA, 2022, pp. 430-435, doi: 10.1109/ISIE51582.2022.9831523.

Warren Liao, T., Zhang, Z., Mount, C.R. Similarity measures for retrieval in case-based reasoning systems. Applied Artificial Intelligence 12:267-288, 1998.

Watson, I., An introduction to Case-based reasoning. Progress in Case-Based Reasoning. LNAI #1020. Pp 3-16, 1996.

Wesseling, C.G., D. Karssenberg, P.A. Burrough, and W.P.A. Van Deursen. (1996). Integrating dynamic environmental models in GIS: the development of a dynamic modelling language. Transactions in GIS, 1(1), 40-48.