INSTITUTO
UNIVERSITÁRIO
DE LISBOA

A Data Quality Framework for Sensing Data

Diogo António Faia Fernandes

Master in Data Science

Supervisor:
Ph.D, Elsa Alexandre Cabral da Rocha Cardoso, Associate Professor,
Department of Information Science and Technology at Iscte-IUL

Ph.D, José Eduardo de Mendonça Tomás Barateiro, Invited Assistant Professor,
Department of Information Science and Technology at Iscte-IUL and Assistant Professor, University of Algarve

May, 2026

BUSINESS SCHOOL

TECNOLOGIAS E ARQUITETURA

Department of Quantitative Methods for Management and Economics

Department of Information Science and Technology

A Data Quality Framework for Sensing Data

Diogo António Faia Fernandes

Master in Data Science

Supervisor:
Ph.D, Elsa Alexandre Cabral da Rocha Cardoso, Associate Professor,
Department of Information Science and Technology at Iscte-IUL

Ph.D, José Eduardo de Mendonça Tomás Barateiro, Invited Assistant Professor,
Department of Information Science and Technology at Iscte-IUL and Assistant Professor, University of Algarve

May, 2026

In memory of my grandparents and my uncle, who are no longer here, but who daily give me strength, protect me, and light my way.

To Marta, my darling and sweetest wife.

Acknowledgments

First, I thank my wife, who is undoubtedly one of my sources of strength. The world needs people like you, Marta.

Secondly, I thank my supervisors Elsa Cardoso and José Barateiro, who helped me complete this research. Without a doubt, this work would not have been possible without their help.

I must also thank my main master's colleagues, Tiago and Nuno, who accompanied me so much throughout these four semesters. Learning together was fun and motivating.

To my family, mother, father and sister, who even from afar always managed to motivate me to reach the end.

It was undoubtedly an extremely important cycle for my development.

Resumo

A crescente integração e utilização de sensores em sistemas de monitorização de estruturas (*Structural Health Monitoring*) tem levado à produção massiva de dados. No entanto, a sua fiabilidade pode ser comprometida por falhas nos sensores, problemas de transmissão, condições ambientais adversas ou deficiências nos processos de aquisição ou transmissão de dados. Neste contexto, a qualidade dos dados utilizados assume um papel crítico, sendo necessária a sua avaliação sistemática para garantir que são adequados ao seu propósito e suportam decisões seguras e informadas.

Esta dissertação propõe o desenvolvimento de uma framework para avaliação da qualidade de dados provenientes de sensores, com aplicação real numa barragem em Portugal. A framework foi concebida para atuar na fase de aquisição de dados sensoriais, após o registo dos mesmos em sistema de informação. Como caso prático de implementação, foram utilizadas séries temporais provenientes de sensores de um instrumento fio de prumo. A solução foi implementada utilizando Python e permite avaliar automaticamente o Profiling dos dados, duas dimensões fundamentais de qualidade de dados – Timeliness e Completeness – e posterior avaliação comparando com thresholds, e a produção de visualizações para todos os passos mencionados.

Palavras-chave: Qualidade de Dados, Gestão de Dados, Redes de Sensores, Falhas em Sensores, Análise Estrutural

Abstract

The increasing integration and use of sensors in Structural Health Monitoring systems has led to the massive production of data. However, the reliability of this data can be compromised by sensor failures, transmission issues, adverse environmental conditions, or deficiencies in acquisition and transmission processes. In this context, the quality of the data becomes critical, and systematic evaluation is essential to ensure that the data is fit for purpose and supports safe and informed decision-making.

This dissertation proposes the development of a framework for evaluating the quality of data collected from sensors, with a real-world application in a dam located in Portugal. The framework was designed to operate during the data acquisition phase, immediately after sensor measurements are recorded and stored. As a practical implementation example, time series data from a plumbline sensor was used.

The solution was developed in Python and enables the automated assessment of data profiling, two fundamental Data Quality dimension – Completeness and Timeliness – followed by threshold-based evaluation and the generation of visualisations for each of these steps.

Keywords: Data Quality, Data Management, Sensor Networks, Sensor Faults, Structural Health Monitoring

Contents

Chapter 1 1

Introduction 1

1.1. Context and Motivation *1*

1.2. Research Questions *2*

1.3. Document Structure *2*

Chapter 2 4

Background 4

2.1. IoT and Data Quality *4*

2.2. Taxonomy of Faults and Anomalies in Sensors *4*

2.3. DQ Frameworks *5*

2.4. Structural Health Monitoring *6*

Chapter 3 7

Related Work 7

3.1. PRISMA *7*

3.1.1. Prisma Extraction Flow Overview 7

3.1.2. Scopus Database Query Interpretation 8

3.1.3. Inclusion and Exclusion Criteria 9

3.2. State of the art *11*

3.2.1. Data Quality and Data Processing from a general, foundational, or systematic perspective 11

3.2.2. Conceptualization and Taxonomy of Dimensions 12

3.2.3. Algorithmic and Statistical Detection Methods 15

3.2.4. Fixing Error methods 16

3.3. Research Gaps *17*

Chapter 4 19

Design and Implementation of the DQ Framework 19

4.1. Chapter Overview *19*

4.2. What is a Data Quality Framework *19*

4.3. Methodological Approach *21*

4.3.1. Architecture design and DQ checks definition 21

4.3.2. Modular development 21

4.3.3. Tests with real sensor data 21

4.3.4. Functional validation of outputs and indicators 21

4.4. Framework Architecture *22*

4.4.1. Management and Policy Layer 22

4.4.2. Data Acquisition Layer 22
4.4.3. Application Layer 23
4.4.4. Visualisation and Reporting Layer 24
4.4.5. Feedback Layer 24
4.5. Chapter Summary *25*

Chapter 5 26

Case Study and Practical Application 26
5.1. Chapter Overview *26*
5.2. Framework application based on the defined Layers *26*
5.2.1. Management and Policy Layer 26
5.2.2. Data Acquisition Layer 27
5.2.3. Application Layer 27
5.2.4. Visualisation and Reporting Layer 41
5.2.5. Feedback Layer 41
5.3. Chapter Summary *41*

Chapter 6 43

Results Discussion 43
6.1. Chapter Overview *43*
6.2. Results *43*
6.3. Results Discussion *55*
6.3.1. Profiling 55
6.3.2. Completeness DQ Dimension 56
6.3.3. Timeliness DQ Dimension 56
6.3.4. Monthly Radial and Radial displacements 58
6.3.5. Feedback Layer outputs analysis 59
6.3.6. Manual vs Automatic acquisition 59
6.4. Chapter Summary *60*

Chapter 7 61

Conclusion 61
7.1. Summary of Results *61*
7.2. Conclusions Regarding the Research Questions *62*
7.3. Contributions to Research and Practice *63*
7.4. Critical Analysis *64*
7.5. Limitations Identified 66
7.6. Future Work *66*

Bibliography 68

Appendix A – Studies found in Literature Review 72

Appendix B – List of files shared by LNEC 75

Appendix C – Configuration Files 76

Appendix D – Application Layer 77

Appendix E – Results 78

Appendix F – Management and Politics Layer Document Template (author proposal) 79

Appendix G – Management and Politics Layer Document Proposal 80

Tables Index

Table 1. Scopus Query Decomposition 9
Table 2. Inclusion Criteria 9
Table 3. Exclusion Criteria 10
Table 4. Studies focused mainly in DQ Frameworks 10
Table 5. Studies focused mainly in data fault management within SHM and IoT 11
Table 6. Articles included 11
Table 7. Articles addressing foundational and systematic DQ perspectives 12
Table 8. Articles conceptualizing DQ taxonomies and dimensions 13
Table 9. Articles conceptualizing specific DQ taxonomies and dimensions 15
Table 10. Articles focusing on anomaly and error detection methods 16
Table 11. Articles focusing on error fixing and data imputation methods 17
Table 12. Common fields between both reading files 27
Table 13. Data directory structure for each execution of the Application Layer 32
Table 14. csv file header for ydata-Profiling csv output 33
Table 15. List of Completeness Data Quality Checks implemented 34
Table 16. List of Timeliness Data Quality Checks implemented 37
Table 17. Threshold Validation Alert Categories 39
Table 18. Examples of Application Layer Execution Command Line 40
Table 19. Ydata-Profiling .csv output for automatic reading mode 45
Table 20. Csv output with year-month distribution values for automatic reading mode – year 2024 46
Table 21. Data Quality Checks .csv output file for automatic read mode – Completeness DQ Dimension 48
Table 22. hours_with_multiple_records_abs sample 49
Table 23. redudant_records_hours_abs sample 49
Table 24. Detailed observations 49
Table 25. Data Quality Checks .csv output file for automatic read mode – Timeliness DQ Dimension 50

Figures Index

Figure 1. PRISMA flow diagram, illustrating the research strategy..... 8
Figure 2. High level Framework modules schema..... 22
Figure 3. Detailed framework modules implementation 26
Figure 4. All input files under instrument input data directory 28
Figure 5. Log file and DQ Checks output 29
Figure 6. Entire project directory structure 30
Figure 7. Output directories for the execution 44
Figure 8. Different Profiling modules outputs 44
Figure 9. Ydata-Profiling output for DIA variable in automatic read mode (.html)..... 45
Figure 10. Year-month distribution visualization for automatic reading mode – year 202446
Figure 11. DQ Checks Outputs in output directory..... 47
Figure 12. Threshold Validator outputs 51
Figure 13. Threshold Validator Reporting output for automatic reading mode – Completeness DQ Dimension (Absolute DQ Check type) 52
Figure 14. Displacement visuals..... 53
Figure 15. Monthly Average Radial Displacement for automatic reading mode 54
Figure 16. Monthly Average Tangential Displacement for automatic reading mode 54
Figure 17. Feedback layer files output..... 55

Abbreviations

DQ – Data Quality

DQ Check – Data Quality Check

DQ Dimension – Data Quality Dimension

IoT – Internet of Things

LNEC – Portuguese National Civil Engineering Laboratory

PDCA – Plan Do Check Act

PRISMA – Preferred Reporting Items for Systematic Reviews and Meta-Analyses

SHM – Structural Health Monitoring

VoI – Value of Information

WSN – Wireless Sensor Network

Chapter 1

Introduction

1.1. Context and Motivation

Civil infrastructure, such as dams, bridges, and tunnels, forms an important part of the backbone of modern society. However, much of this infrastructure is aging and subject to continuous environmental stress, material degradation, and operational loads. Historically, assessing the safety of these structures relied on periodic manual inspections, which are costly, time-consuming, and often incapable of detecting sudden structural shifts (Fu et al., 2019).

To address this, the field of Structural Health Monitoring (SHM) has increasingly adopted the Internet of Things (IoT) and Wireless Sensor Networks (WSNs). By embedding low-cost, interconnected sensors – such as accelerometers, strain gauges, and piezometers – directly into the infrastructure, engineers can transition from reactive maintenance to real-time, data-driven, predictive monitoring (Nguyen et al., 2023).

While IoT provides the necessary hardware architecture for continuous observation, it introduces a critical software and analytical challenge: Data Quality (DQ). Sensors in SHM systems are typically deployed in harsh, inaccessible outdoor environments. They are exposed to for example to extreme weather, electromagnetic interference, and battery depletion (Venkatesan et al., 2017). Consequently, these WSNs inevitably generate "dirty data" – streams plagued by missing values, sudden spikes, continuous drift, and noise (Fu et al., 2019).

This reality forms the primary motivation for this master thesis. In the context of critical infrastructure, poor data quality is not merely an IT inconvenience but yes, a profound safety risk. If a data stream from a structure contains unchecked anomalies, it could lead to catastrophic false negative scenarios (failing to detect an imminent collapse) or costly false positive scenarios (triggering unnecessary emergency evacuations based on a faulty sensor spike) (Fang et al., 2025).

Therefore, before any advanced structural analysis, machine learning, or predictive maintenance algorithms can be reliably applied, the spatiotemporal integrity of the raw sensor data must be guaranteed (Nguyen et al., 2023). There is a pressing need for robust, automated Data Quality Management Frameworks capable of detecting anomalies, differentiating true structural events from sensor faults, and intelligently imputing missing data. This thesis is motivated by the necessity to bridge the gap between unreliable physical IoT sensors and the high-fidelity data required for dependable SHM.

1.2. Research Questions

This dissertation focuses on how a Data Quality Framework can be designed and applied into a network sensing data applying it in a real scenario. To guide the design and evaluation of the framework, a total of three research questions were created:

RQ1 – What are the Data Quality problems that affect sensors in civil engineering structures?

RQ2 – What are the relevant Data Quality Dimensions for sensor data in civil engineering structures?

RQ3 – What are the most common scenarios for this type of problems?

1.3. Document Structure

This dissertation is structured in seven chapters: Chapter 1 (Introduction) where the context motivation and the research questions for this dissertation are presented. Chapter 2, Background, serving as a context for the mainly concepts presented in this dissertation. After, Chapter 3 (Related Work) is presented with the Systematic Literature Review, encompassing previous investigations done in IoT and Data Quality subjects, focused on IoT data errors and Data Quality Dimensions. Chapter 4 (Design and Implementation of the DQ Framework) describes the investigation done on how to design and build a Data Quality framework accordingly DAMA (DAMA International, 2017a) and ISO 8000-61 principles (King & Schwarzenbach, 2020) , demonstrating how a Data Quality Framework can be built to fit into a network sensing data to perform Data Quality checks and evaluate results against defined thresholds, reporting all needed info at the end. Chapter 5 (Case Study and Practical Application) demonstrates how to apply the designed Data Quality framework in a real scenario using a plumbline dataset shared by the Portuguese National Civil Engineering Laboratory (LNEC), from a Portuguese dam. Chapter 6 (Results Discussion) presents the results output from the case study from Chapter 5 and finally Chapter 7 (Conclusion) presents the conclusion from this dissertation and also its limitations, how it

answers the different research questions, the contributions to LNEC and ending with open work proposals for future investigations.

Chapter 2

Background

2.1. IoT and Data Quality

The rapid proliferation of IoT has fundamentally transformed how the physical world is monitored and managed. IoT networks, particularly WSNs, are composed of interconnected, resource-constrained devices that continuously collect and transmit data to centralized or edge computing systems for processing (Teh et al., 2020) (Karkouch et al., 2016). The core value of an IoT ecosystem lies not only in the data it generates but also in the physical nature of these sensors and their deployment in harsh, remote environments make them highly susceptible to errors.

Data Quality in IoT extends beyond traditional database management definitions. While conventional DQ focuses on static data, high quality data in IoT must account for high velocity, volume, and streaming characteristics (Mansouri et al., 2021). IoT Data Quality is multidimensional, typically categorized into intrinsic, contextual, representational, and accessibility dimensions (Liu et al., 2020). In the context of sensor networks, it is possible to identify critical Data Quality dimensions like:

- Accuracy**:** The proximity of the sensor reading to the true physical value.
- Completeness**:** The absence of missing values or data gaps over a continuous temporal window.
- Timeliness**:** The delay between data generation and its availability for processing.
- Consistency**:** The logical correlation between multiple sensors monitoring the same physical phenomenon.

When these dimensions are compromised, the resulting "dirty data" can lead to erroneous analyses, false alarms, and poor decision-making. As IoT systems increasingly drive automated systems, addressing these DQ challenges at the source and network levels has become an important research focus (Karkouch et al., 2016).

2.2. Taxonomy of Faults and Anomalies in Sensors

To effectively manage data quality, it is essential to understand the root causes of data corruption. Errors in WSNs are generally from two primary sources: the physical degradation of the sensor node (hardware source) and communication network failures (Karkouch et al., 2015; Nguyen et al., 2023).

A structured taxonomy of sensor faults is critical for developing targeted anomaly detection algorithms. These faults can be of several distinct patterns (Fu et al., 2019):

- Bias (or Offset): A constant error added to the true signal. The sensor consistently reports values that are higher or lower than the actual physical phenomenon by a fixed amount.
- Drift: A time-dependent degradation where the sensor reading gradually diverges from the true value. This is highly common in environmental monitoring due to factors like chemical depletion, battery drain, or biofouling.
- Gain (Scaling): The rate of change in the sensor's measurement is multiplied by an erroneous constant, causing the amplitude of the data series to distort.
- Precision Degradation (Noise): The variance of the sensor readings increases significantly, often due to low battery or electromagnetic interference.
- Spikes (Short-term Anomalies): Sudden, isolated extreme values that deviate massively from the expected temporal pattern before returning to normal.
- Missing Data (Data Loss): Complete loss of signal for varying durations, typically caused by routing failures, power loss, or physical destruction of the node.

It is highly important to distinguish between a faulty sensor (an error in the measurement device) and a true anomaly (an unusual but real physical event occurring in the environment) (Nguyen et al., 2023).

2.3. DQ Frameworks

To mitigate the risks associated with poor data quality, researchers have proposed various Data Quality Management Frameworks tailored for IoT. These frameworks aim to systematically evaluate, monitor, and improve the veracity of data streams before they reach end-user applications (Ehrlinger & Wöß, 2022).

Modern DQ frameworks generally adopt a multi-layered approach consisting of edge-level and cloud-level processing (Karkouch et al., 2018):

- Data Profiling and Measurement: Incoming sensor streams are continuously evaluated against predefined DQ metrics (such as calculating the percentage of missing values or detecting high-variance windows).
- Anomaly Detection: Machine learning and statistical algorithms (such as Deep Reinforcement Learning, predictive modelling) are employed to identify outliers, spikes, and drifts in real-time.

- Data Cleaning and Imputation: Once bad data is identified, the framework must repair it. (Kuemper et al., 2018) propose *ValidIoT*, a spatial-temporal correlation to not only flag errors but also interpolate and reconstruct missing or corrupted data points using neighbour sensor values.

These model-driven architectures ensure that the data fed into predictive analytics or decision-support systems maintains a baseline level of trustworthiness, isolating application logic from the underlying hardware unreliability.

2.4. Structural Health Monitoring

A critical domain where the convergence of IoT and Data Quality is highly visible is SHM, that involves the continuous observation of civil infrastructure – like dams, bridges, and tunnels – using a set of embedded sensors (e.g., accelerometers, strain gauges, piezometers) to track structural integrity and material deformation over time (Fu et al., 2019). In this area the deployment of IoT sensor networks allows for real-time data-driven approaches to infrastructure safety that drastically improves upon traditional manual inspections (C. Feng et al., 2025). However, dams and bridges are often located in severe environmental conditions, making their sensor networks highly prone to fault, like in section. A sensor drift or a series of gross errors (Fang et al., 2025) in a dam's deformation monitoring series, could either trigger a catastrophic false evacuation alarm or, conversely, mask a true structural failure and so applying robust Data Quality Frameworks to SHM is not merely an IT optimization but yes a public safety necessity. Before any structural assessment models (like predictive maintenance or AI-integrated damage detection) can be trusted, the underlying spatiotemporal integrity of the sensor data must be verified and cleaned (C. Feng et al., 2025; Fu et al., 2019).

CHAPTER 3

Related Work

3.1. PRISMA

This section presents the research that was done helping find an answer to the research questions presented before, presenting all the inclusion and exclusion criteria for studies found there, considering the context of this SLR.

The framework used was PRSIMA, from which the indicated guidelines were followed (Page et al., 2021). Also, the flow diagram used to visually document the entire search process is presented in the next section, describing the process of including articles with their respective sources, screening, eligibility and selection of articles based on inclusion and exclusion criteria, taking into account the context of the SLR (Page et al., 2021).

3.1.1. Prisma Extraction Flow Overview

The study selection process was conducted in multiple phases to ensure a rigorous and transparent filtering of the literature, ultimately isolating the most relevant studies for the master thesis goal.

The initial search query executed in the Scopus database returned a total of 129 articles. Before applying all the inclusion and exclusion criteria, an immediate practical screening was performed to assess the availability and publication status of the retrieved records. During this initial phase, 7 articles were immediately excluded for the following reasons: 2 articles lacked a Digital Object Identifier (DOI), 3 articles were entirely inaccessible (full-text not available), and 2 articles were pre-prints or not yet officially published. This practical filtering reduced the corpus to 122 articles.

In the second phase, a primary screening was conducted by reading the titles and abstracts of the remaining 122 articles. The end goal of this phase was to quickly identify and remove literature that was fundamentally misaligned with the research scope. Based on the abstract screening, 94 articles were excluded, leaving 28 articles that demonstrated potential relevance to the study.

In the final phase, a comprehensive full-text review was performed on the remaining 28 articles. During this stage, the predefined Inclusion Criteria (IC) and Exclusion Criteria (EC) were strictly applied to each article. Articles were evaluated on their specific data architecture and their theoretical or mathematical applicability. Following this rigorous evaluation, 9 articles were excluded for failing to satisfy the specific IC/EC logical pathways.

As a result of this systematic filtering process, a total of 19 highly relevant articles was selected that constitute the core literature analysed in this review. The full list of articles is included in the Appendix A.

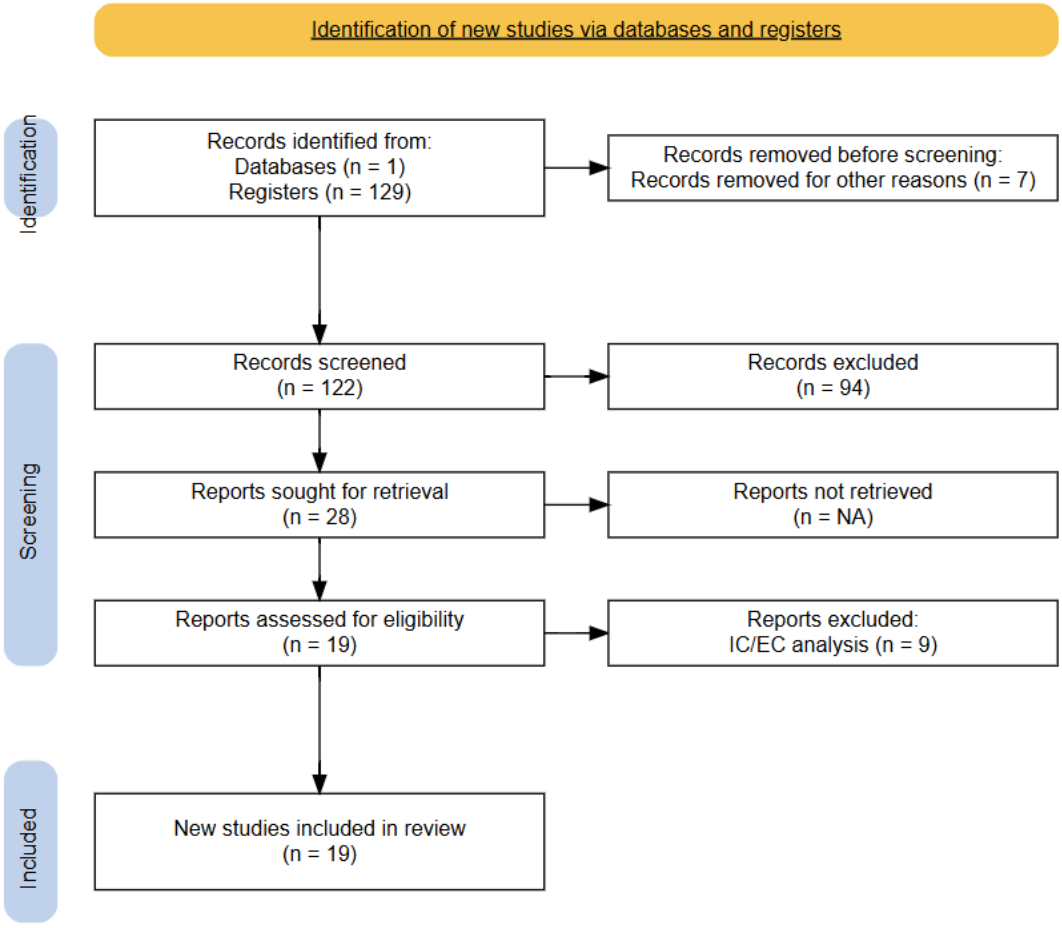


Figure 1. PRISMA flow diagram, illustrating the research strategy

3.1.2. Scopus Database Query Interpretation

For the literature search, Scopus was selected as the primary database. This choice is justified by its status as one of the largest and most comprehensive databases of peer-reviewed literature globally as it also provides an extensive multidisciplinary coverage that ensures a wide capture of relevant studies.

The search query in this database should include terms such as "*Data Quality*", "*data governance*", "*iot*", "*sensor*", "*sensing*", "*dam*", "*bridge*", "*structure*" or "*structural*", considering a 10-year period of publications between 2014 and 2024. Only articles written in English were considered. The query was executed on 15/01/2025 and all articles were extracted on that date.

Below the final resulting and optimized query:

((TITLE ("data quality" OR "data governance" OR "data quality framework" OR "data quality dimensions") OR KEY ("data quality" OR "data governance" OR "data quality framework" OR "data quality dimensions")) AND (TITLE (iot OR sensor OR sensing) OR KEY (iot OR sensor OR sensing)) AND TITLE-ABS-KEY (dam OR bridge OR structure OR structural OR "civil engineering" OR smh OR "structural health monitoring" OR damage) AND NOT (spatial, AND lidar, AND imag)) AND PUBYEAR > 2013 AND PUBYEAR < 2025 AND (LIMIT-TO (DOCTYPE , "ar") OR LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "re")) AND (LIMIT-TO (LANGUAGE , "English"))*

Below in Table 1 the decomposition of the query:

Table 1. Scopus Query Decomposition

Concept	Data Quality or data governance
Population	iot OR sensor OR sensing
Context	Specific infrastructures (dam OR bridge OR structure OR structural)
Document Type	Article, Conference Paper, Reviews
Publication Period	2014 to 2024 (10 Years)
Articles Language	English

3.1.3. Inclusion and Exclusion Criteria

After the first screened trough only Abstract section for each one of the articles, a more detailed and carefully classification was made based on the selection criteria (inclusion/exclusion) presented in Table 2 and Table 3. This step becomes important at this point in the research because it is necessary to ensure that no research is rejected and that those that are included, are in accordance with the research questions defined previously.

Table 2. Inclusion Criteria

ID	Category	Criterion Description
IC1	Focus on Data Quality and Data Management	Studies proposing foundational Data Quality tools, taxonomies, Big Data models, or metrics (including domain-agnostic frameworks)
IC2	Focus on Data Quality and Signal Processing	Studies investigating methods to clean, process, or extract features from sensor data (e.g., imputation, outlier detection, wavelet transforms, signal processing)
IC3	Civil /Structural Health Monitoring study domain	Studies applied to Civil Engineering and Structural Health Monitoring
IC4	Water domain	Any study applied to water management, hydraulic structures, or hydrology
IC5	Sensors & IoT domain	Studies focusing on sensor data quality, WSNs, or IoT monitoring systems.

Table 3. Exclusion Criteria

ID	Category	Criterion Description
EC1	Image data processing / Visual data	Studies exclusively using computer vision, images, or remote sensing
EC2	Irrelevant Domains	Studies strictly applied to non-engineering domains (e.g., Healthcare) unless they propose a domain-independent, foundational Data Quality framework
EC3	Peripheral Focus	Studies where Data Quality, data pre-processing, or signal anomaly detection is not a central subject of the investigation
EC4	Publication Type	English as not the main language.

The selected literature must sit at the intersection of data quality management, structural or aquatic monitoring, and time-series sensor data processing.

To achieve this, explicit Inclusion Criteria (IC) and Exclusion Criteria (EC) were defined. The inclusion criteria focus on capturing domain-agnostic DQ theories, algorithmic signal processing methods, and specific applications in SHM and water management.

Conversely, the exclusion criteria were designed to filter out incompatible data architectures and/or domains entirely disconnected from physical infrastructures, unless they provide a foundational framework.

A boolean logic approach was implemented to ensure that distinct types of necessary literature were captured. The 22 initially retrieved articles were evaluated through three distinct logical pathways, resulting in a final 19 included articles. The first pathway follows the logic to meet IC1 and explicitly propose a generic/domain-independent framework, bypassing EC2 (irrelevant domains) helping build a comprehensive DQ framework that requires an understanding of overarching Big Data taxonomies. Here studies included are applied to generic or business databases, providing the fundamental theoretical building blocks for data quality dimensions.

The detailed studies included are described in the table below:

Table 4. Studies focused mainly in DQ Frameworks

Study	Inclusion criteria reason
(Miller et al., 2024)	Provides a domain-agnostic, hierarchical data model for DQ terminology
(Batini et al., 2015)	Establishes foundational theory on how DQ evolves within Big Data environments

The second pathway meets a logic that captures studies dealing directly with sensor data cleaning, imputation, and fault management within SHM and industrial IoT environments, being so focused on SHM & IoT. The list of included articles is described below:

Table 5. Studies focused mainly in data fault management within SHM and IoT

Study	Inclusion criteria reason
(X. Feng et al., 2013)	Wavelet transforms for SHM signal processing
(Fu et al., 2019)	SHM sensor fault management
(Richter et al., 2024)	Pre-processing for fiber optic strain sensing
(Giordano et al., 2023)	Value of Information (VoI) in SHM
(Nishio, 2017)	Bridge strain sensor data reliability
(Karkouch et al., 2015)	IoT outlier detection
(Guggilam & Sundaram, 2024)	Multivariate IoT data imputation
(Weiß & Vogel-Heuser, 2021)	Data completeness metric for industrial sensors
(Kim et al., 2019)	Smart connected product DQ operations
(Tawakuli & Engel, 2022)	IoT pre-processing pipelines
(Adhikari et al., 2021)	Iterative imputation for high missing gaps
(Byabazaire et al., 2022)	IoT trust and data quality
(Tang et al., 2015)	Fuzzy C-means imputation for time-series sensors

The last pathway is focused on Water Domain, as dams operate in complex aquatic environments, understanding sensor behaviour and operational data quality in water contexts is a primary constraint of this thesis. This pathway ensures that any study dealing with water telemetry or aquatic sensor environments is preserved. The list of articles included is below:

Table 6. Articles included

Study	Inclusion criteria reason
(Gleeson et al., 2023)	Drinking water distribution DQ frameworks.
(Frontera-Bergas et al., 2024)	Embedded DQ assessment in flash flood monitoring.
(Bhatti et al., 2019)	Systematic sensor data analysis for water flow/canal operations.
(Venkatesan et al., 2017)	The physical impact of marine environments (biofouling) on sensor time-series data integrity.

All bibliographic references were stored using the Mendley platform.

3.2. State of the art

3.2.1. Data Quality and Data Processing from a general, foundational, or systematic perspective

This section explores the literature that addresses DQ and data processing through overarching frameworks, architectures, and systematic pipelines. Before addressing specific errors in sensor data, it is crucial to understand how data is managed, valued, and structured at a macro level, particularly as environments scale into Big Data and IoT ecosystems.

(Batini et al., 2015) provide a foundational investigation into how traditional DQ issues have evolved with the advent of Big Data, highlighting the shift from structured relational databases to semi-structured and sensor-driven datasets. Building upon this, (Byabazaire et al., 2022) investigate the intrinsic relationship between IoT Big Data models, Data Quality, and the resulting "Trust" users can place in automated systems. In industrial and operational contexts, (Kim et al., 2019) extend traditional DQ management strategies specifically to accommodate smart connected products, emphasizing that continuous monitoring requires dynamic quality lifecycles.

When applying these systematic concepts to physical infrastructure and water domains, literature demonstrates a strong need for standardized frameworks. (Gleeson et al., 2023) propose a dedicated DQ assessment framework tailored for drinking water distribution systems, evaluating time-series datasets before they are used for actionable insights. Similarly, (Tawakuli & Engel, 2022) advocate for normalizing the design phase of data pre-processing pipelines, ensuring that data transformations in IoT are reproducible and standardized regardless of the specific sensor type. The operational value of these frameworks is quantified by (Giordano et al., 2023), who investigate the Value of Information (VoI) in SHM, mathematically demonstrating how imperfect data quality impacts decision-making. Finally, (Bhatti et al., 2019) provide an operational perspective on systematic data processing by analysing sensor telemetry for canal operations, where systematic data flow is critical for performance management.

Table 7. Articles addressing foundational and systematic DQ perspectives

Study	Domain	Investigation brief description
(Batini et al., 2015)	Big Data	The evolution of data quality paradigms from traditional databases to semi-structured and sensor-driven Big Data
(Kim et al., 2019)	IoT / Industry	Extending traditional data quality management strategies for smart connected sensor operations
(Tawakuli & Engel, 2022)	IoT / Data Pipelines	Standardizing and normalizing the design phase of data pre-processing pipelines for continuous IoT data
(Byabazaire et al., 2022)	IoT / Trust	The theoretical intersection and modeling of IoT Big Data, data quality, and system trust
(Gleeson et al., 2023)	Water Networks	Development of a comprehensive DQ assessment framework specifically for water distribution time-series datasets
(Giordano et al., 2023)	SHM	Quantifying the Value of Information (VoI) in structural monitoring when dealing with imperfect data quality
(Bhatti et al., 2019)	Water	Systematic sensor data telemetry analysis for forecasting and operating water flow in canals

3.2.2. Conceptualization and Taxonomy of Dimensions

To effectively measure data quality, the literature establishes specific taxonomies and dimensions (e.g., Accuracy, Completeness, Reliability). Understanding how these dimensions

are defined, quantified, and visualized in the context of continuous monitoring is essential when defining a Data Quality Framework.

To effectively measure and manage data quality, the literature establishes that "quality" cannot be treated as a single variable, but it must be broken down into specific, quantifiable characteristics known as Data Quality Dimensions. (Batini et al., 2015) note that while traditional database management relied on static dimensions, the advent of Big Data and continuous IoT sensing has forced these dimensions to evolve, requiring dynamic, continuous evaluation.

To standardize the disparate nomenclature found across various scientific disciplines, (Miller et al., 2024) propose a comprehensive, domain-agnostic hierarchical framework that aggregates both established and emerging DQ dimensions, providing a standardized taxonomy that can be adapted to highly specific use cases.

Translating these theoretical dimensions into practical metrics, (Weiß & Vogel-Heuser, 2021) focus specifically on the dimension of "Completeness," developing a specialized metric and visualization method for multidimensional condition monitoring sensors in automated systems.

In physical infrastructure environments, the conceptualization of DQ often overlaps with the physical reliability of the sensor itself. (Nishio, 2017) evaluates the long-term stability and reliability dimensions of fiber-optic strain data acquired from concrete bridges, emphasizing how environmental deterioration impacts the physical spectra of the data. Similarly, (Venkatesan et al., 2017) investigate the physical dimension of data integrity by studying how biofouling affects the reliability of sensor-mounted moored observatory systems in marine environments. Together, these studies highlight that for physical infrastructure like dams, data quality dimensions must account for both digital metrics (like completeness) and physical degradation (like biofouling).

Table 8. Articles conceptualizing DQ taxonomies and dimensions

Study	Domain	Investigation brief description
(Miller et al., 2024)	Foundational	Creating a hierarchical framework to standardize current and new data quality dimensions and terminology
(Weiß & Vogel-Heuser, 2021)	Industrial Sensors	Developing a quantifiable metric and visualization tool specifically for the DQ dimension of "completeness"
(Nishio, 2017)	SHM	Evaluating the long-term reliability, stability, and data quality degradation of fiber-optic strain sensors
(Venkatesan et al., 2017)	Water	Investigating how physical marine biofouling impacts sensor reliability and the integrity of time-series data

Within the reviewed literature applied to sensor networks and infrastructure, the investigation of these dimensions primarily focuses on the following key areas:

Completeness refers to the degree to which a dataset contains all required values without missing chronological gaps. In continuous automated monitoring, missing data is one of the most frequent anomalies due to network disconnections or power failures. (Weiß & Vogel-Heuser, 2021) specifically target this dimension, developing a quantifiable metric and visualization tool for "Completeness" in multi-dimensional condition monitoring sensors. Their work emphasizes that before any structural analysis can be performed, the system must first mathematically verify the completeness of the time-series continuum.

For physical infrastructure and aquatic environments, the dimensions of reliability and accuracy are inextricably linked to the physical state of the hardware. Data cannot be mathematically accurate if the sensor itself is compromised.

(Nishio, 2017) investigates the long-term stability and reliability dimensions of fiber-optic strain data acquired from concrete bridges. The study demonstrates how the environmental deterioration of the sensor's physical spectra directly degrades the accuracy of the recorded data over the structure's lifecycle.

In aquatic environments, (Venkatesan et al., 2017) investigate the physical dimension of data integrity by studying how marine biofouling affects sensor-mounted moored observatory systems. Their findings highlight that in water-based monitoring, assessing data reliability requires accounting for the physical degradation of the submerged sensors.

As sensor networks scale into massive IoT architectures, the "veracity" (Trust) of the data becomes a critical dimension. (Byabazaire et al., 2022) investigate the intrinsic relationship between IoT Big Data models, data quality dimensions, and system trust. They propose that trust is a meta-dimension; it is the ultimate metric that determines whether an automated system or a human operator can confidently utilize the sensor data for critical decision-making.

Timeliness evaluates whether the data is available at the time it is needed for operational decision-making. While often overlooked in static post-processing, it is vital in real-time monitoring. (Bhatti et al., 2019), in their analysis of sensor telemetry for water canal operations, highlight latency, data processing delays, and transmission periods as critical dimensions. For active water management, delayed data is effectively low-quality data, as it cannot be used to execute time-sensitive hydraulic control operations.

Table 9. Articles conceptualizing specific DQ taxonomies and dimensions

Study	Domain	Primary DQ Dimensions Investigated	Investigation brief description
(Miller et al., 2024)	Foundational	Taxonomy, All dimensions	Creating a hierarchical framework to standardize current and new data quality dimensions and terminology
(Batini et al., 2015)	Big Data / IoT	Evolution of Dimensions	Investigating how traditional dimensions (Accuracy, Completeness) evolve in semi-structured Big Data contexts
(Weiß & Vogel-Heuser, 2021)	Industrial Sensors	Completeness	Developing a quantifiable metric and visualization tool specifically for the DQ dimension of completeness
(Nishio, 2017)	SHM (Bridges)	Reliability, Stability, Accuracy	Evaluating the long-term reliability and data quality degradation of fiber-optic strain sensors on infrastructure
(Venkatesan et al., 2017)	Water (Oceanic)	Reliability, Integrity	Investigating how physical aquatic biofouling impacts sensor reliability and the integrity of time-series data
(Byabazaire et al., 2022)	IoT	Trust, Veracity	The theoretical intersection and modelling of IoT Big Data models, data quality, and user/system trust
(Bhatti et al., 2019)	Water (Canals)	Timeliness, Latency	Highlighting telemetry latency and processing delays as critical operational dimensions in water flow management

3.2.3. Algorithmic and Statistical Detection Methods

Before errors in a time-series dataset can be fixed, they must be accurately detected. This section synthesizes the literature focusing on the algorithms and statistical methods used to identify anomalies, outliers, and sensor faults.

(Karkouch et al., 2015) provide a comprehensive review of data outlier detection techniques in IoT environments, highlighting how the discovery of statistical anomalies is the first crucial step in data cleaning. In the context of SHM, (Fu et al., 2019) propose advanced sensor fault management techniques for wireless smart sensor networks, utilizing distributed computing algorithms to detect faults locally before erroneous data corrupts the centralized system.

Signal processing techniques are frequently employed for detection in structural engineering. (X. Feng et al., 2013) investigate the use of continuous wavelet transforms to process fiber optic sensor data, successfully detecting anomalies and physical damage by analysing frequency shifts in the time-series signal. (Richter et al., 2024) further advance this domain by exploring data pre-processing and anomaly detection methods specifically for distributed fiber optic strain sensing. Finally, in the water monitoring domain, (Frontera-Bergas et al., 2024) demonstrate the practical implementation of detection algorithms by designing urban flash flood monitoring stations equipped with embedded, real-time data quality assessment and anomaly detection protocols.

Table 10. Articles focusing on anomaly and error detection methods

Study	Domain	Investigation brief description
(Karkouch et al., 2015)	IoT	Reviewing techniques for outlier discovery and statistical anomaly detection in continuous IoT streams
(X. Feng et al., 2013)	SHM	Utilizing wavelet transforms on sensor data for signal processing and structural anomaly/damage detection.
(Fu et al., 2019)	SHM	Designing distributed algorithms for sensor fault detection and management in wireless smart sensor networks.
(Frontera-Bergas et al., 2024)	Water (Floods)	Designing a water monitoring station with embedded firmware algorithms for real-time DQ assessment and fault detection.
(Richter et al., 2024)	SHM	Investigating advanced pre-processing and detection methods for high-resolution distributed strain sensing data.

3.2.4. Fixing Error methods

Once data quality issues are identified corrective methods must be applied to reconstruct the dataset.

(Adhikari et al., 2021) proposes an iterative, ratio-based imputation method specifically designed to handle extreme missing gaps in IoT time-series systems, outperforming standard mean-imputation techniques. Taking a hybrid approach, (Tang et al., 2015) combine fuzzy C-means clustering with genetic algorithms to accurately estimate and impute missing volume data in transportation sensors, proving that evolutionary algorithms can effectively reconstruct missing chronological patterns.

More recently (Guggilam & Sundaram, 2024) introduced a highly complex neural network architecture: a Spatio-Temporal Bidirectional Long Short-Term Memory based Variational Auto-Encoder (ST-Bi-LSTM-VAE). This model effectively captures both spatial correlations between multiple sensors and temporal chronological dependencies to impute missing multivariate IoT data with high precision.

Table 11. Articles focusing on error fixing and data imputation methods

Study	Domain	Investigation brief description
(Tang et al., 2015)	Time-Series Sensors	Integrating fuzzy C-means clustering with genetic algorithms to estimate and fix missing sensor data
(Adhikari et al., 2021)	IoT	Proposing an iterative ratio-based imputation algorithm capable of fixing extreme, continuous missing data gaps
(Guggilam & Sundaram, 2024)	IoT	Developing a deep learning approach (ST-Bi-LSTM-VAE) to impute missing multivariate spatial-temporal sensor data.

3.3. Research Gaps

The comprehensive analysis of the 19 articles offers robust DQ frameworks, they are predominantly designed for either generic, high-speed IoT environments (Kim et al., 2019; Tawakuli & Engel, 2022) or distinctly different water domains, such as drinking water distribution (Gleeson et al., 2023) and flash flood monitoring (Gleeson et al., 2023).

The current literature provides excellent generic models for data quality, but it stops short of providing a practical, deployable solution for a single dimension structural sensors in dam environments. A significant gap exists in the literature for a domain-specific framework that applies foundational DQ architecture directly to the safety-critical and highly regulated environment of hydraulic dams.

The foundational literature successfully defines the taxonomy of Data Quality. Studies by (Miller et al., 2024) and (Batini et al., 2015) studies are focused on the essential dimensions such as Accuracy, Completeness, and Timeliness while (Weiß & Vogel-Heuser, 2021) provide methods to visualize completeness in industrial sensors. However, a major gap remains in how to operationally control these dimensions within a continuous monitoring loop. The literature often treats DQ dimensions as passive metrics to be evaluated after data collection, not accurately reflecting the daily reality of dam monitoring where these dimensions must be actively controlled through automated thresholds near the origin of the data (sensors).

This master's thesis aims to directly bridge these gaps. By utilizing the foundational taxonomies identified in this Systematic Literature Review, this research will propose a Data Quality Framework specifically designed for dam plumblines. The framework will not only establish which DQ dimensions are critical for structural safety – Completeness and Timeliness – but will also provide a systematic method to identify common error scenarios and actively control the quality of the data lifecycle. Ultimately, this will ensure that the displacement data used by engineers to assess dam stability is reliable, complete, and trustworthy.

CHAPTER 4

Design and Implementation of the DQ Framework

4.1. Chapter Overview

This chapter provides a detailed description of the design, development, and implementation process of the framework proposed in this dissertation. Following the analysis presented in Chapter 3, specific needs were identified within the context of sensor network data monitoring.

The chapter begins with a theoretical overview of the concept of a Data Quality framework, followed by a detailed explanation of the methodological approach adopted, the modular system architecture, and finally, the technical implementation of the two selected dimensions: Timeliness and Completeness. The chapter concludes with a description of the outputs generated by the framework, highlighting their value for analysis, interpretation, and future integration into sensor network monitoring pipelines and decision support systems.

Based on the conclusions drawn from the previous literature review chapters, a structured vision was defined regarding how a Data Quality framework can be designed to address the specific challenges related to DQ Dimensions, integrating quality dimensions, specific metrics, and automated processes for assessment and reporting.

4.2. What is a Data Quality Framework

The definition of a Data Quality Framework in the context of sensor networks represents a critical step in ensuring that collected data is reliable and fit for its intended purpose. In environments characterised by heterogeneous, volatile, and high-volume data, Data Quality management becomes essential to safeguard the usefulness and integrity of information flows (Batini et al., 2015). The primary goal of a Data Quality Framework is to guarantee the principle of fitness for use, meaning the data's suitability for its specific purpose (DAMA International, 2017b).

According to (Weiß & Vogel-Heuser, 2021) a Data Quality framework should be established following a management process that, at a high level, is divided into Assessment and Improvement. For example, (Bhatti et al., 2019) propose a six-step approach: Definition, Assessment, Analysis, Control, Implementation, and Improvement.

DAMA, a global non-profit association dedicated to advancing data management practices, defines Data Quality Management in four steps (DAMA International, 2017a):

1 – Develop a governed approach to ensure data is fit for purpose, based on the requirements of data consumers;

2 – Define standards, requirements, and specifications for Data Quality controls, integrating them across the data lifecycle;

3 – Define and implement processes to measure, monitor, and report Data Quality levels;

4 – Identify and promote opportunities for improving Data Quality by optimising processes and systems.

(King & Schwarzenbach, 2020) describe the ISO 8000-61 standard as an international reference model for Data Quality management, structured around the PDCA cycle (Plan-Do-Check-Act), a fundamental principle of quality management. This model defines 20 core processes, organised into four functional areas (Planning, Control, Assurance, and Improvement) and two supporting areas (Data-related Support and Resource Provision). In the planning phase, data requirements, quality strategies, policies, and implementation plans are defined. The control phase ensures that data is created and updated according to these requirements, including monitoring and validation. The assurance phase focuses on measuring Data Quality and evaluating the effectiveness of processes, enabling evidence-based adjustments. Finally, the improvement phase introduces practices such as root cause analysis, data cleansing, and process reinforcement to prevent future non-conformities. The support areas include managing data architecture, security, operations, and data transfer, as well as organisational and skills development. The standard also proposes an evolutionary approach through five capability levels, ranging from basic data processing (Level 1) to full maturity with continuous improvement and innovation (Level 5). This model provides a comprehensive and scalable vision for implementing data governance and quality practices in organisations, promoting a systematic, measurable, and sustainable approach over time.

In the context of sensor networks, a Data Quality framework should account for the specific characteristics of data acquisition throughout the network (Gleeson et al., 2023): Space, where quality can be assessed at the level of individual sensors, the entire network, or any of its components, time where quality can be evaluated for single data points or data streams (typically associated with time-series) and data sources where data is captured by a wide variety of devices (sensors) and may take different forms (digital, analogue, spatial, temporal, or even manual measurements). Quality depends on factors such as sensor type, manufacturer, deployment age, and calibration technique (Byabazaire et al., 2022).

Data Profiling is the process of analysing the content of a database or other data source (Sargiotis, 2024). DAMA International also refers to Data Profiling as a statistical analysis technique used to examine the content, structure, and quality of datasets (DAMA International, 2017c). By using this set of techniques, it becomes possible to identify patterns, outliers, format inconsistencies, null values, and frequency distributions, where both at the level of individual columns and through cross-analysis between columns and tables.

The results of profiling should be interpreted, ideally with the support of data analysts, to confirm business rules, uncover hidden patterns, or assess compliance with predefined requirements for data implementation and usage. Although essential for initial diagnosis, profiling is only the first step in the Data Quality improvement process and should be complemented with process analyses, data lineage evaluations, and root cause identification.

4.3. Methodological Approach

4.3.1. Architecture design and DQ checks definition

Based on the scientific literature review, a general architecture was proposed for a Data Quality framework aimed at the systematic evaluation of Data Quality in sensor networks. This architecture is built upon key components such as the definition of relevant quality dimensions, the use of specific DQ Checks for each dimension, the application of data profiling techniques, and the integration of automated monitoring and reporting mechanisms.

The theoretical framework analysed thus served as the foundation for structuring a modular, scalable solution aligned with best practices in Data Quality management within sensory contexts.

4.3.2. Modular development

With the aim of enabling future integration of new sensors, the framework was developed using a modular architecture, composed of distinct components with individual and independent functionalities. This approach facilitates scalability and reusability, allowing the framework to be adapted to different types of sensors and monitoring contexts.

4.3.3. Tests with real sensor data

Following the development phase, the framework was tested using real data from a SHM environment there by ensuring a practical validation of its applicability. The use of sensor data collected in a real-world context made it possible to assess the framework's performance in detecting Data Quality issues such as acquisition failures, missing values, or anomalous patterns, as well as its ability to present those results effectively, ensuring the relevance and robustness of the proposed solution.

4.3.4. Functional validation of outputs and indicators

As a final step, the outputs generated by the framework were analysed, and the necessary adjustments were made to improve its reliability and ability to deliver trustworthy results. The framework was designed following an iterative and incremental logic, allowing it to function both as a solid foundation for Data Quality assessment and as an evolving artefact. This characteristic ensures that the framework can be continuously adapted to new sensors, requirements, or contexts, guaranteeing its relevance, scalability, and adaptability in the future.

4.4. Framework Architecture

The functional structure of the framework is organised into five interdependent layers, each with specific responsibilities within the Data Quality lifecycle. This layered approach enables different user profiles (from technical to business-oriented) to interact with the system in an easy and effective manner. Each layer represents a distinct phase in the Data Quality management process, ranging from the definition of policies and rules to data collection, assessment, visualisation, and the detailed feedback of all records evaluated for the different Data Quality Checks (DQ Checks). This organisation not only facilitates the maintenance of the framework but also promotes continuous improvement.

The layers are presented and detailed in the following sections, and Figure 2 shows the different framework components:

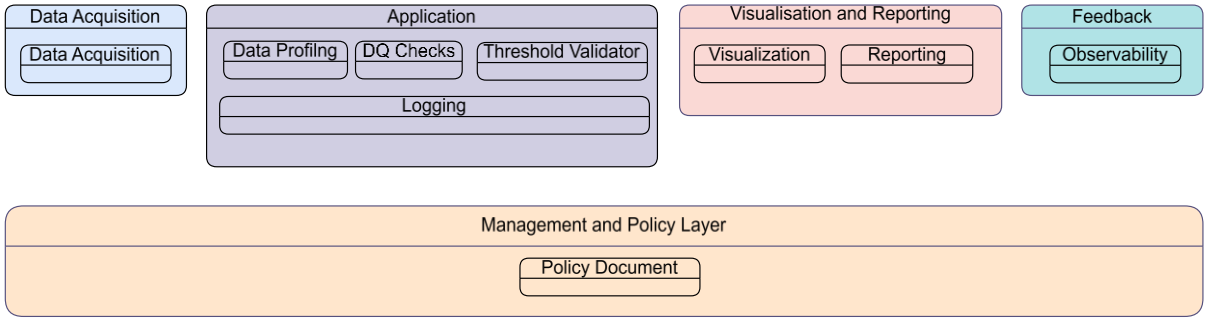


Figure 2. High level Framework modules schema

4.4.1. Management and Policy Layer

The Management and Policy Layer constitute the starting point of the framework, assuming a theoretical, methodical, and iterative role focused on the governance and continuous management of Data Quality. This layer establishes the guiding principles that ensure consistency and implementation of the framework, through the definition of data flows associated with each sensor, acquisition rules (such as data collection frequency or whether the reading process is manual or automatic), and the responsibilities assigned to each stakeholder.

The main output of this layer is a version-controlled document accessible to the entire organisation, in which these policies are formalised, along with the roles and responsibilities of data stewards, continuous improvement plans, technical documentation of the sensors, quality checks per dimension, and the critical fields to be monitored. This document serves as a reference, essential to ensure transparency, auditability, and the strategic alignment of the framework with the organisation's objectives.

4.4.2. Data Acquisition Layer

This layer is responsible for collecting data from the sensors of different instruments used within an environment.

The main functions of this layer are essentially to ensure that data is collected systematically and consistently placed in the predefined directories, as agreed upon by all stakeholders, always in the same manner and without failures.

At this stage, file-level validations may be applied to ensure that the expected frequency of data file reception is respected, as well as the correct file format.

4.4.3. Application Layer

This layer constitutes the core of the framework, where the main Data Quality assessment processes are executed in a sequential manner. The layer begins with the application of Data Profiling techniques, enabling an understanding of the data structure and distribution. Subsequently, Data Quality checks are applied per dimension, and the results are compared against predefined thresholds using the Threshold Validator, allowing for the automatic identification of deviations, anomalies, and levels of criticality.

As output, this layer produces interpretable reports and visualisations, facilitating the interpretation of results by analysts and data stewards. The execution of this layer depends on a set of essential configurations, stored in parameterisable files by end users, where acceptable limits for each DQ Check, target columns, and the expected behaviour of each dimension per instrument can be defined. This ensures the framework's flexibility and adaptability to different contexts, sensors, and operational requirements.

4.4.3.1. Profiling Module

This layer is primarily responsible for generating reports in order to help data analysts to identify data patterns, outliers and to get a set of a basic quantitative metrics of the data observed for each one of the desired columns. When necessary, data cleansing procedures might be manually applied to ensure that, in the Application Layer, only clean and reliable data is processed. However, if such transformations occur, it is crucial to ensure that all applied functions are documented and reviewed within the Feedback Layer, as this will be essential to prevent similar issues from recurring in future data acquisitions. This feedback will be particularly valuable for the Data Stewards responsible for the various sensors and instruments, as it will help prevent future errors during the data acquisition phase, thus avoiding repetitive data cleaning processes.

4.4.3.2. Data Quality Checks Module

The DQ Checks Layer forms the core of the framework, being responsible for executing Data Quality checks for each considered dimension: Timeliness and Completeness. This layer automatically applies a set of DQ Checks to the data, with the primary objective of identifying anomalies, irregular patterns, or failures that may compromise the integrity and usefulness of the data.

Each check is parameterisable and can be adapted to the specific context of the sensor in use. The outputs of this layer include dimension-specific reports in *.csv* format. By centralising Data Quality assessment within this layer, the framework ensures consistency in the application of rules across each execution.

4.4.3.3. Threshold Validator Module

The main function of this layer is to validate the results obtained from the Data Quality checks by comparing them against predefined thresholds for each dimension. This layer acts as a control and alert mechanism, allowing results to be classified into categories such as "*OK*", "*WARNING*", "*ALERT*", or "*NO THRESHOLD*", based on the comparison between observed values and configured thresholds.

This approach enables an objective assessment of the data, allowing for its contextual interpretation according to defined parameters. The Threshold Validation Layer is thus essential for supporting decision-making, automatically highlighting situations that require attention and contributing to the continuous improvement of Data Quality.

4.4.3.4. Logging Module

This layer accompanies the entire execution of the application layer, logging every step performed and allowing direct consultation through a *.log* generated file.

4.4.4. Visualisation and Reporting Layer

The Visualisation and Reporting Layer is responsible for transforming the results of the Data Quality checks and threshold validator into accessible, and interpretable information for the various stakeholders involved in the data management process, particularly data analysts.

Functioning as the output layer, it focuses on generating visual reports in *.png* image format, enabling the monitoring of Data Quality status for each execution. This layer plays a crucial role in facilitating communication between stakeholders, as it allows the tracking of Data Quality evolution, the identification of alerts and deviations, and the implementation of corrective actions when necessary.

4.4.5. Feedback Layer

The Feedback Layer serve as the feedback mechanism for all evaluated records during the DQ Checks step, accelerating any future steps needed for analysis. The primary objective of this layer is to present the detected anomalies and help on the correction processes, ensuring that identified issues are traced back to their root cause and properly addressed to the assigned teams or people.

In doing so, this layer reinforces the iterative and adaptive nature of the system, promoting a sustained cycle of continuous improvement in Data Quality.

4.5. Chapter Summary

This chapter demonstrated how the Data Quality assessment framework was designed and implemented to meet the specific requirements of data acquisition in sensor networks. Through a modular and parameterisable approach, the developed solution enables the execution of profiling techniques, the automatic evaluation of multiple quality dimensions, and ultimately the generation of outputs that support decision-making. This makes it a framework with clear advantages: accessibility for end users, code modularity that allows the integration of new sensors, result logging, and ease of integration with visualisation tools or operational platforms.

The framework developed in this dissertation proves to be practically applicable, establishing itself as an essential tool for ensuring data reliability through systematic and structured evaluation. This solution serves as the foundation for the practical case study explored in the following chapters, focused on SHM.

Chapter 5

Case Study and Practical Application

5.1. Chapter Overview

This chapter aims to present the practical application of the developed framework, validating its effectiveness in detecting Data Quality issues in a real-world environment. To this end, data from plumb line sensors installed at a Portuguese dam were used, allowing the framework's capabilities to be tested.

The structure of this chapter follows the processes and objectives defined in Chapter 4.

5.2. Framework application based on the defined Layers

Figure 3 below shows the different modules for each one of the framework layers.

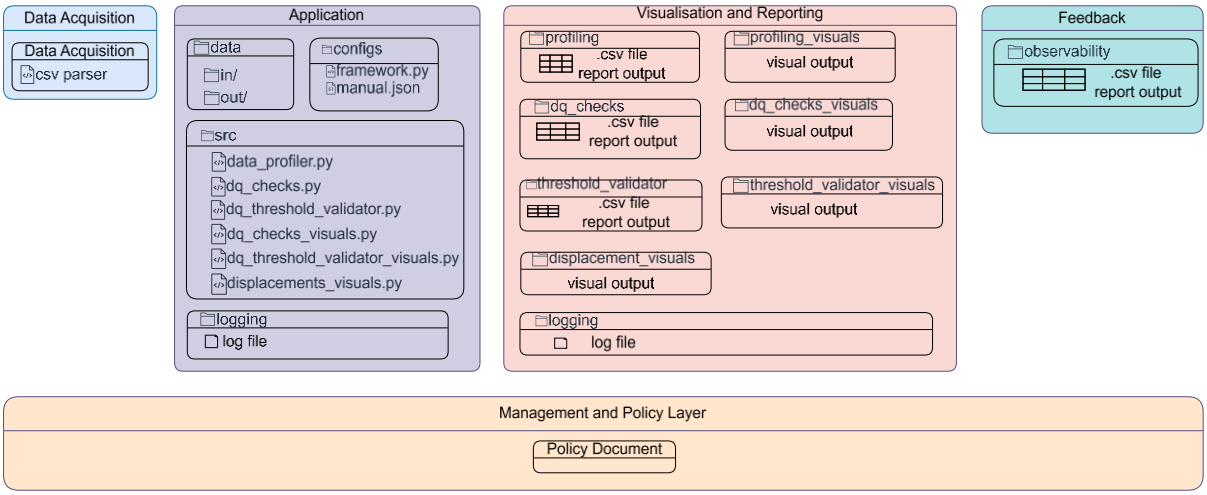


Figure 3. Detailed framework modules implementation

5.2.1. Management and Policy Layer

In this layer, a document was produced (included in the Appendix G) which plays a fundamental role in the formalisation of the framework. It defines its purpose and scope, presents a high-level description of the sensors considered, and outlines the roles and responsibilities assigned to each stakeholder (individual or department). The data flow throughout the different stages is also described, as well as the functional requirements of the application layer. The document also allowed for the registration of essential characteristics for both reading modes, such as data acquisition frequency and relevant columns, as well as the definition of the quality metrics to be implemented and their corresponding thresholds. Given the experimental context of this dissertation, some fields were completed in a generic manner and should be further detailed in a real-world scenario. Nevertheless, the document served as a support throughout the entire

implementation phase of the framework, acting as a foundation for its configuration, parameterisation, and alignment with the objectives of the dissertation.

5.2.2. Data Acquisition Layer

The data used in this study were provided by LNEC and refer to raw manual and automatic readings for four plumb line sensors. Each sensor has two files, one for each type of reading. The reading types are identified by the first three characters of the filename: *RAD* for automatic readings and *RMD* for manual readings.

In total, eight files were obtained for two reading types: *automatic* and *manual*. All files are in *.txt* format, with tab-separated values, in accordance with the specifications defined in the Governance and Policy document created in the Governance and Policy Layer of the framework.

The list of all files that were shared by LNEC are in the Appendix B.

Both types of reading modes have in common the following data fields, as described in Table 12:

Table 12. Common fields between both reading files

Field	Description	Data Type
ANO	Year of the reading	Integer
MES	Month of the reading	Integer
DIA	Day of the reading	Integer
HORA	Hour of the reading	Integer
MINUTO	Minute of the reading	Integer
DESLOCRADIALABS	Absolute value of the radial displacement of the plumb line position	Decimal
DESLOCTANGABS	Absolute value of the tangential displacement of the plumb line position	Decimal

5.2.3. Application Layer

5.2.3.1. Execution Flow

The execution of the framework begins by running a command in the command line. The data must already be placed in the appropriate input directory.

Below, Figure 4 shows all the available files to be evaluated:

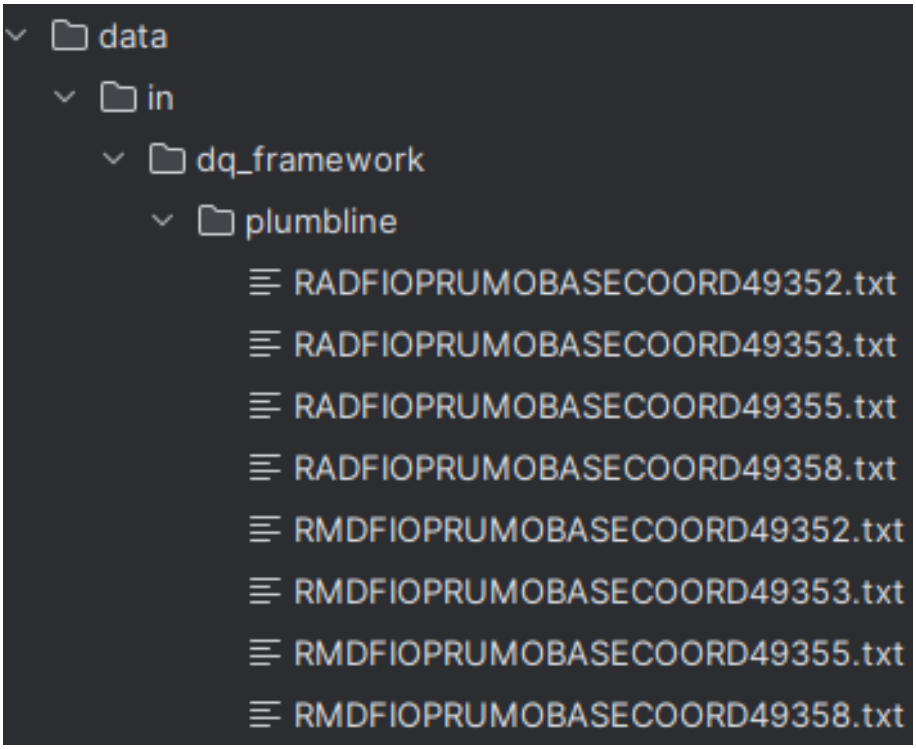


Figure 4. All input files under instrument input data directory

To run the framework for “plumbline” instrument for “2025” “automatic” readings for “49358” sensor, the next command can be used: *.\src\dq_framework\dq_framework_exec.py --instrument plumbline --mode all --id_list 49358 --year_list 2024*. For the context of this dissertation, data was anonymously used avoiding public identification of the sensor used.

Next, the application layer retrieves the data for the specified instrument passed as a command-line argument, filtering it by the indicated list of years, if provided. The profiling process is then initiated, generating the corresponding reports.

Afterwards, the DQ Checks module evaluates the data for the selected reading mode and sensor IDs, also defined in the command-line arguments, producing the respective output files. These output files are then consumed by the Threshold Validator module, which applies threshold evaluation formulas and generates another set of outputs.

Finally, the output files from both the DQ Checks and Threshold Validator modules are processed by the visualisation module, which produces the corresponding graphical outputs.

Another set of visualisations is created under *displacements* folder – *displacements_visuals.py*. This set of visualisations was created specifically for this dissertation context, because both variables DESLOCRADIALABS and DESLOCTANGABS suffer seasonal fluctuations, and these visualisations can and should be combined with the DQ Checks and Threshold Validator outputs. There are two types of visualisations for each one of the target variables: Monthly Absolute values, in a violin plot, and Monthly Average Values, combining absolute values (in a violin plot) with the monthly average value.

All generated outputs are saved in an output directory named after the instrument, followed by a timestamped subfolder in *YYYYMMDDHHMISS* format, making the results available to support decision-making processes. Below Figure 5 shows an example output of the framework, where it is possible to see all the DQ Checks outputs (*.csv* reports and visuals) and the log file.

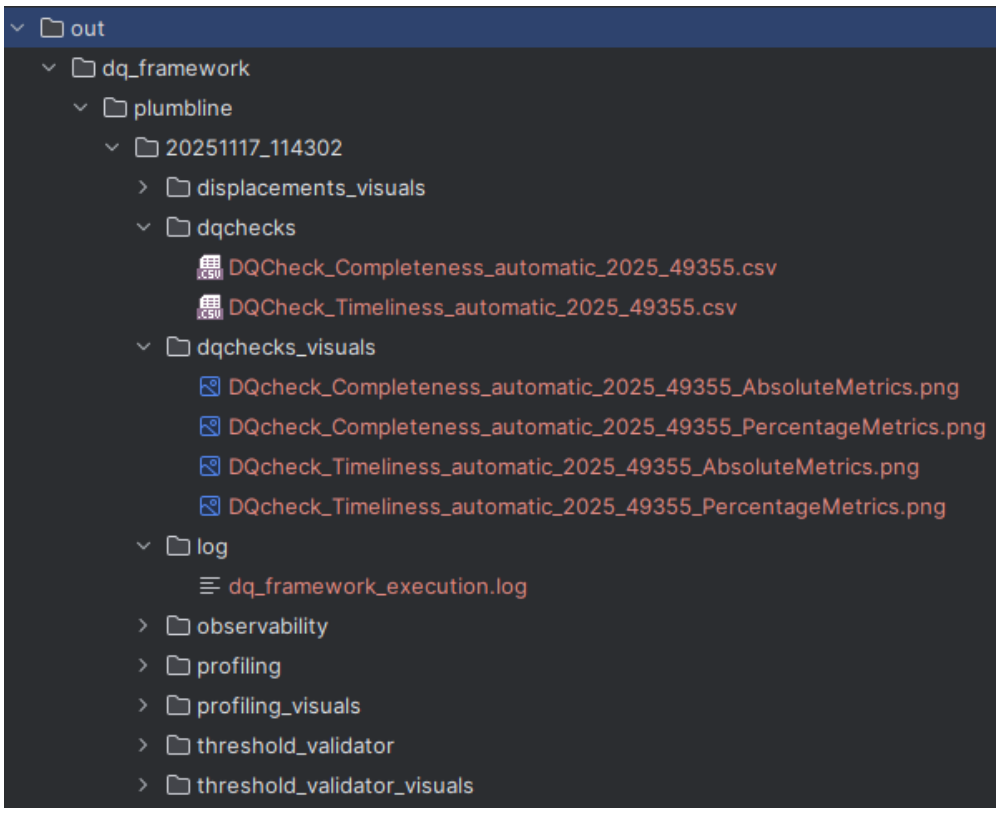


Figure 5. Log file and DQ Checks output

The entire application layer and the way its different modules interact is described below. All modules in this layer were fully developed in Python.

The technical structure is based on the following folder organisation:

- *configs* – for configuration files;
- *data* – for data storage;
- *src* – for the individual development of each module that makes up the framework.

Below Figure 6 shows the entire framework directory project where it is possible to see the different directories and some of the files/scripts included in.

The *src* layer constitutes the operational core of the framework, aggregating execution scripts into functional modules and utility components that support the entire Data Quality assessment lifecycle. The structure was designed to be modular and scalable, allowing for the creation of an independent subdirectory for each new instrument to be integrated into the framework.

Within each instrument directory, the following specific execution modules can be found: *profiling*, *dq_checks*, *threshold_validator*, *visuals*, as well as the main execution script for the respective instrument.

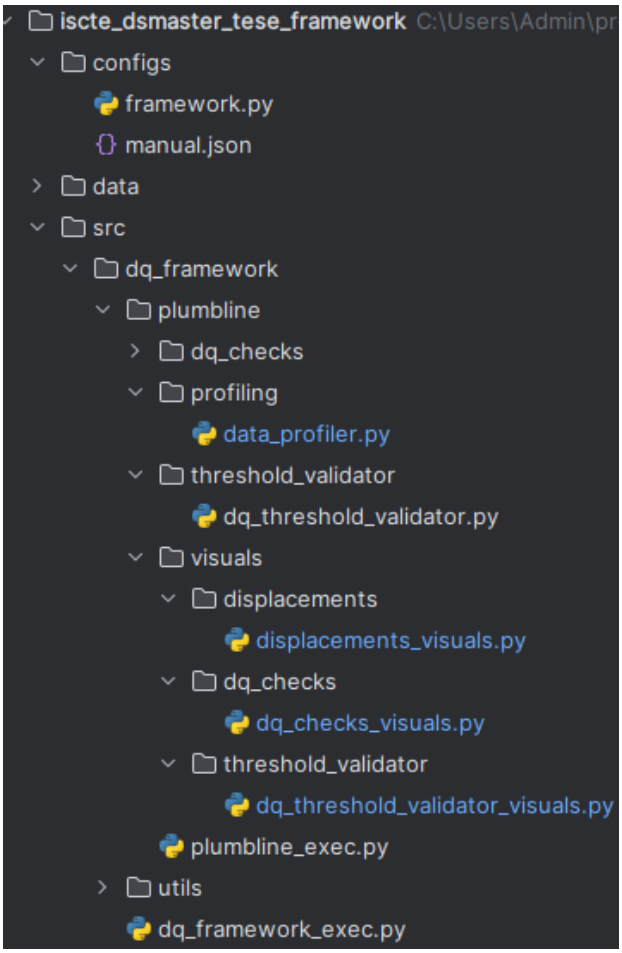


Figure 6. Entire project directory structure

5.2.3.2. Configs

The configuration management of the framework is centralised in *<root>/configs/*, which contains the necessary files for parameterising the operation of the application layer. This structure includes two main files: *manual.json* and *framework.py*.

manual.json file is responsible for storing instrument-specific configurations, allowing the definition, for each sensor, of parameters such as the types of files to read, applicable metrics, threshold values, columns to consider for profiling, and other variables relevant to the framework operation.

The *framework.py* file contains the global configurations of the framework, including base directories, folder names, environment variables, and other application parameters that apply across all sensors. This organisation promotes flexibility and scalability, making it easier to adapt the framework to different operational contexts without needing to open the project in a development environment configuration is achieved simply by editing these files directly. This ensures a clear separation between code logic and parameterisation, contributing to the framework maintainability and reusability.

The application layer uses a set of global variables that support the framework execution. These variables are defined in the file located at *<root>/configs/framework.py*. All variables in this file must be named in uppercase, clearly indicating that they are global across all used modules.

For the sensor data evaluated in this study, it was also necessary to define a specific set of global variables, which were also configured in this file.

As a good practice, for future instruments, the variables should always follow the same naming convention, with the instrument name appearing first.

In the Appendix C of this dissertation, it is possible to get all variables defined for the different *framework.py* and *manual.json* configuration files.

5.2.3.3. Data

For data storage, the main data directory — *data* — is located at *<root>/data* and is divided into two subdirectories: *in* and *out*. The directory structure of the framework was designed primarily to ensure clarity regarding the type of files.

The *in/* directory serves as the general input repository, where all data required for running the framework must be stored. This directory contains a specific subdirectory — *dq_framework* — which is then subdivided by instrument allowing the input data to be isolated by sensor type.

The *out/* directory centralises all outputs generated by the framework. Within it, the *dq_framework* subdirectory stores the results from the application layer, organised first by instrument and then by execution timestamp. Each execution creates a new folder named with the timestamp in the format *YYYYMMDD_HHMISS*, containing all relevant outputs, which are further separated into folders with the designations as describe in Table 13 This structure enables historical analysis of results, allowing for easy comparison between executions. Table 13, describes each output folder and respective description:

Table 13. Data directory structure for each execution of the Application Layer

Folder Name	Description
Displacements_visuals	Displacements visualisations
dq_checks	Data Quality check results
dq_check_report_visuals	Visualisations of DQ checks
log	Log files
profiling	Profiling results
profiling_visuals	Profiling visualisations
dq_threshold_validator	Threshold validation results
dq_threshold_validator_visuals	Visualisations of threshold validations

5.2.3.4. Log

This directory stores the log file – *dq_framework_execution.log* – for the entire execution of the Application Layer. All steps executed store the information here, to be later consulted by any person, helping not only understand the entire execution but also debugging any error, if any during the execution.

5.2.3.5. Profiling Module

The data profiling assessment is performed by the script located at *<root>/src/plumbline/profiling/data_profiler.py*. This module includes two components, the first being the generation of two aggregate visualisations: one by year, and another by year and month.

To enable these outputs, some transformations were applied to the data. First, the column *MES* was decoded into a new column *MES_STR*, and then a final column *YEAR_MONTH_STR* was created by concatenating the values from *ANO* and *MES_STR*. This allows the generation of charts showing the ordered distribution of record counts for each year and month.

This component produces two types of output files: *.csv* and *.png*: *.csv* files contain the aggregated results and their respective values while *.png* files represent the visualisations of those results.

The different tables in Appendix D, show the output file naming convention for the different output files. Also in the Appendix D, different tables describe the *.csv* file for year-based and year-month distribution.

The second component involves the use of *ydata-profiling*[1] library: a Python library that automatically generates exploratory, helping on automating and standardizing report generation with key statistics and visualizations. This component primarily relies on the variable *profiling_columns* from the manual configuration file, which defines the following list of columns present in both automatic and manual reading modes: "*ANO*", "*MES*", "*DIA*", "*HORA*", "*MINUTO*", "*DESLOCRADIALABS*", and "*DESLOCTANGABS*".

As output, this module generates two types of files: *.csv* and *.html*. These reports include metrics such as value distribution, cardinality, missing values, and data types. The *.html* report is a complete interactive report, including built-in visualisations. File naming convention for this report can be checked in the Appendix D.

The output *.csv* file provides a descriptive summary of statistical metrics for the columns used in the profiling process. Thus, for each column (i.e., each field analysed during the profiling process), a set of metrics is presented along with their corresponding observed values. The output metrics are described in Table 14 Table 14. csv file header for ydata-Profiling csv outputbelow:

Table 14. csv file header for ydata-Profiling csv output

Metric	Description
count	Total number of records for the variable
mean	Mean of the observed values for the variable
std	Standard deviation of the observed values for the variable
min	Minimum observed value for the variable
25%	First quartile (Q1) – 25% of the values are less than or equal to this value
50%	Median (Q2) – Represents the central value of the distribution; 50% of the data is above and 50% below this value
75%	Third quartile (Q3) – 75% of the values are less than or equal to this value
max	Maximum observed value for the variable

On the other hand, the *.html* file generates a comprehensive and interactive report, which, when opened in a browser, allows users to explore basic statistics for the selected fields.

5.2.3.6. DQ Checks Module

The Data Quality assessment module — *dq_checks.py* — implements the algorithms and parameterisable rules for the quality dimensions.

[1] https://docs.profiling.ydata.ai/latest/

To operationalise the Data Quality checks, the manual configuration file includes the following settings for both Data Quality dimensions. These fields are defined separately for the automatic and manual reading modes, allowing the criteria to be tailored to the operational characteristics of each sensor type.

Manual configuration fields and respective values for Completeness and Timeliness dimensions reading modes are defined in the Appendix D.

For the implementation of the Data Quality checks, the fields *DATA, DATA_DAY, DATA_HOUR*, and *DATA_MINUTE* are calculated internally when the reading file is loaded into memory. These fields are derived from the existing columns *ANO*, *MES*, and *DIA*, and are combined into a date-formatted structure, enabling operations based on *datetime* format values.

It is also during this step, for each one of the DQ Checks implemented, the outputs for the Feedback Layer are produced.

Table 15 and Table 16 next, present all the DQ Checks implemented respectively for both Completeness and Timeliness DQ Dimensions.

5.2.3.6.1. Completeness

The objective of this DQ Dimension is to detect data collection failures by the sensors, loss of expected cadence, or anomalous interruptions in data acquisition.

The full table with all the Data Quality checks implemented for this DQ Dimension and the corresponding reading mode in which each check was applied is defined below:

Table 15. List of Completeness Data Quality Checks implemented

DQ Check	Description	Output Type	Purpose	Reading Mode
days_with_multiple_records_abs	Number of days with more than one record	Absolute value	Detects days with redundant data entries	Automatic Manual
days_with_multiple_records_prctg	Percentage of days with more than one record	Percentage	Measures the proportion of days containing multiple records, indicating potential redundancy.	Automatic Manual
hours_with_multiple_records_abs	Number of hours with more than one record	Absolute value	Identifies redundant readings within the same hour.	Automatic

hours_with_multiple_records_prctg	Percentage of hours with more than one record	Percentage	Quantifies how frequently duplicate hourly readings occur.	Automatic
minutes_with_multiple_records_abs	Number of minutes with more than one record	Absolute value	Detects excessive data frequency or duplication within the same minute.	Automatic
minutes_with_multiple_records_prctg	Percentage of minutes with more than one record	Percentage	Indicates potential oversampling or repeated readings in short time intervals.	Automatic
missing_data_day_abs	Number of days with missing data	Absolute value	Measures the total number of days without records.	Automatic Manual
missing_data_day_prctg	Percentage of days with missing data	Percentage	Quantifies the proportion of missing daily records relative to the expected total.	Automatic Manual
missing_data_hour_abs	Number of missing hourly records	Absolute value	Counts the number of expected but absent hourly readings.	Automatic
missing_data_hour_prctg	Percentage of missing hourly records	Percentage	Evaluates hourly completeness as a percentage of total expected readings.	Automatic
missing_data_minute_abs	Number of missing minute-level records	Absolute value	Detects missing data points at minute-level granularity.	Automatic

missing_data_minute_prctg	Percentage of missing minute-level records	Percentage	Expresses the completeness of minute-level data relative to the expected recording frequency.	Automatic
missing_DESLOCRADIALABS_abs	Number of missing values in the DESLOCRADIALABS column	Absolute value	Detects missing radial displacement readings.	Automatic Manual
missing_DESLOCRADIALABS_prctg	Percentage of missing DESLOCRADIALABS values	Percentage	Quantifies the completeness of the radial displacement series.	Automatic Manual
missing_DESLOCTANGABS_abs	Number of missing values in the DESLOCTANGABS column	Absolute value	Detects missing tangential displacement readings.	Automatic Manual
missing_DESLOCTANGABS_prctg	Percentage of missing DESLOCTANGABS values	Percentage	Measures tangential displacement data completeness relative to the expected set.	Automatic Manual
redundant_records_day_abs	Number of redundant records detected per day	Absolute value	Identifies duplicate records on the same day.	Automatic Manual
redundant_records_hours_abs	Number of redundant records detected per hour	Absolute value	Detects redundant hourly records.	Automatic
redundant_records_minutes_abs	Number of redundant records detected per minute	Absolute value	Identifies duplicated minute-level readings.	Automatic

DQ Check column represents the Data Quality Check implemented and *Description* its description. The columns *Output Type* represent the type of output values (“Absolute” for non-percentage values and “Percentage” for percentage values), *Purpose* describing DQ Check goal in the framework (what it is supposed to measure) and *Reading Mode* to mention to which type of reading this DQ Check is being applied for.

5.2.3.6.2. Timeliness

This DQ Dimension evaluates whether the data was collected according to the expected frequency, identifying delays, failures, or temporal irregularities in data acquisition by a sensor of the instrument.

The table below presents the implemented Data Quality checks, along with their description, type of output value, objective, and the reading mode in which they were applied.

Table 16. List of Timeliness Data Quality Checks implemented

DQ Check	Description	Output Type	Purpose	Reading Mode
data_day_density_abs	Record density per day	Absolute value	Evaluates the average daily recording frequency.	Automatic Manual
data_hour_density_abs	Record density per hour	Absolute value	Assesses hourly regularity of date, particularly relevant in automatic mode.	Automatic
data_minute_density_abs	Record density per minute	Absolute value	Measures data granularity per minute.	Automatic
days_apart_previous_record_count_abs	Number of times two consecutive records are separated by days	Absolute value	Quantifies interruptions in consecutive days.	Automatic Manual
days_apart_previous_record_proportion_prctg	Proportion of records separated by days	Percentage	Measures the proportion of gaps longer than one day between consecutive records.	Automatic Manual
hours_apart_previous_record_count_abs	Number of times consecutive records are separated by hours	Absolute value	Identifies hourly gaps in the data stream for automatic sensors.	Automatic
hours_apart_previous_record_proportion_prctg	Proportion of records separated by hours	Percentage	Detects hourly discontinuities as a percentage of total records.	Automatic
interval_days_abs	Average interval between records in days	Absolute value	Measures the average spacing between records.	Automatic Manual
interval_days_prctg	Proportion of acceptable intervals between records in days	Percentage	Evaluates the compliance of daily intervals with the expected recording cadence.	Automatic
interval_hours_abs	Average interval between records in hours	Absolute value	Useful to measure average spacing of automatic readings.	Automatic

interval_hours_prctg	Proportion of acceptable intervals between records in hours	Percentage	Measures compliance of hourly intervals with the expected cadence.	Automatic
interval_minutes_abs	Average interval between records in minutes	Absolute value	Measures short-term variations in the spacing between records.	Automatic
interval_minutes_prctg	Proportion of acceptable intervals between records in minutes	Percentage	Measures compliance of minute-level intervals with the expected cadence.	Automatic
max_gap_days_abs	Largest interval (gap) between consecutive records in days	Absolute value	Detects the longest daily data gap.	Manual
max_gap_hours_abs	Largest interval between consecutive records in hours	Absolute value	Identifies the longest automatic reading failure in terms of hours.	Automatic
max_gap_minutes_abs	Largest interval between consecutive records in minutes	Absolute value	Measures the largest temporal granularity gap in the dataset.	Automatic
minutes_apart_previous_record_count_abs	Number of times consecutive records are separated by minutes	Absolute value	Counts sub-hour reading gaps (e.g., high-frequency sensors).	Automatic
minutes_apart_previous_record_proportion_prctg	Proportion of minute-based separations between consecutive records	Percentage	Percentage of records with minute-level discontinuities; useful for precision sensors.	Automatic

As output, this module generates, within the output directory of the respective sensor and execution, a *.csv* file containing the calculated values for each DQ Check.

5.2.3.7. Threshold Validator

The threshold validator module — *dq_threshold_validator.py* — implements the algorithms and rules used to determine how far a DQ Check value can deviate from the expected.

This module is responsible for executing threshold validations for each calculated DQ Check. It relies on the parameters defined for the instrument in the *manual.json* configuration file, with the *limit_factor* parameter being particularly important, as it plays a crucial role in the threshold validation step of the framework. It works as a sensitivity control parameter in the classification of DQ Check results.

Its purpose is to introduce a tolerance margin between the defined acceptable threshold value (*OK*) and the critical limit above which a result is classified as *ALERT*. Thus, when a DQ Check value slightly exceeds the threshold but remains within the range defined by *threshold×limit_factor*, the result is classified as *WARNING*, representing a situation that deserves attention but is not yet critical.

This behaviour allows the framework's strictness to be adjusted according to the characteristics of the data and the monitoring context, making the validation process more flexible and adaptable.

In practical terms, lower values of *limit_factor* make the system more sensitive, triggering alerts with small variations while higher values introduce greater tolerance before a DQ Check is classified as critical.

As a result of these evaluations, status alerts are generated for each DQ Check: *NO THRESHOLD, WARNING, OK, ALERT*.

Alert generation is defined separately for automatic and manual readings, although the same logic is applied to both. Below the pseudocode that can be applied to generate alerts:

```
if threshold is None:
    classified_result[dq_check] = ("NO THRESHOLD",
value)
else:
    if value <= threshold:
        classified_result[dq_check] = ("OK", value)
    elif value <= threshold * limit_factor:
        classified_result[dq_check] = ("WARNING",
value)
    else:
classified_result[dq_check] = ("ALERT", value)
```

Table 17 below presents the possible alert classifications generated by the threshold validation module. Each DQ Check is evaluated against its defined threshold, and based on the result, an appropriate status is assigned.

Table 17. Threshold Validation Alert Categories

Alert Type	Description
NO THRESHOLD	DQ Check without a defined threshold
OK	When the value calculated in the Threshold Validator module is lesser or equal to the defined threshold value
WARNING	When the value calculated in the Threshold Validator module is lesser or equal than the calculation of defined threshold×limit_factor
ALERT	When the value calculated in the Threshold Validator module is value greater than the calculation of defined threshold×limit_factor

To operationalise the threshold evaluation, threshold values must be defined by DQ dimension, reading mode in *manual.json* configuration file. Similarly to *dq_checks* block in the same configuration file, the fields are defined separately for the *automatic* and *manual* reading modes, allowing the criteria to be tailored to the operational characteristics of each sensor type.

The manual configuration file for the different threshold values is configured accordingly the tables specified in the Appendix D.

As output, this module generates, within the output directory of the respective sensor and execution, a *.csv* file containing the calculated values for each DQ Check.

The output file follows the naming convention and header, and both can be checked in the Appendix D.

5.2.3.8. Utils

The framework also includes a set of utility functions, centralised in the module *utils.py*, located at *src/dq_framework/utils*. This module provides logging mechanisms, directory validation, and helper functions to support the execution of the various modules within the application layer.

5.2.3.9. Execution

Finally, to execute the framework, all components of the Application Layer are run together by simply navigating to the project root directory via the command line and executing the script *dq_framework_exec.py* via command line. All the available parameters are defined in the Appendix C.

Below is a translation of some examples of framework execution:

Table 18. Examples of Application Layer Execution Command Line

Command Line Example	Description
python .\src\dq_framework\dq_framework_exec.py --instrument plumbline --mode all --id_list 00000 --year_list 2024	Executes the framework for the plumbline instrument, for all reading modes of sensor 00000, processing only the year 2024.
python .\src\dq_framework\dq_framework_exec.py --instrument plumbline --mode automatic --id_list all --year_list all	Executes the framework for the plumbline instrument, for automatic readings of all sensors, across all years.

If a required parameter is missing or the input value is incorrect, the code will terminate with an error, and the entire execution log will be saved to the log file.

5.2.4. Visualisation and Reporting Layer

This module aims to translate the results from the DQ Check and Threshold Validator modules into visual outputs, supporting data analysts in interpreting the information and providing feedback to the respective Data Stewards.

For both modules, visualisations are generated based on the files read, with a clear distinction between absolute DQ Check and percentage DQ Check, to simplify interpretation due to the differences in scale.

Also, this module generates four additional visualisations related with both numerical variables *DESLOCRADIALABS* and *DESLOCTANGABS*. The script is located under *displacements/* and is responsible to only generate these visuals.

The output file naming convention follows the formats defined in the Appendix D.

5.2.5. Feedback Layer

This layer represents the final step of the framework, that can be seen as both the closure of a cycle (the Data Quality evaluation with all the steps already executed) and the beginning of a new one – the possibility to evaluate the DQ Checks outputs in a detailed analysis, providing all the information needed for further investigation. Within the scope of this dissertation, the framework was defined and executed, and all results were shared exclusively with LNEC.

The detailed data is available for four specific DQ Checks: *hours_with_multiple_records_abs*, *redundant_records_hours_abs*, *days_apart_previous_record_count_abs*, *max_gap_days_abs*.

The results provided for this layer, are of two output types when applicable: first (and always available), all the records evaluated during the DQ Checks evaluation step (detailed version) and in second, the aggregation of the results. As output, during the execution of DQ Checks Module, the directory *observability* is created under the *timestamp output directory* and the different *.csv* files for each one of the DQ Checks are created, following the naming convention defined in the Appendix D.

The full code is publicly available in a github repository file and can be checked online[2].

5.3. Chapter Summary

In summary, this chapter demonstrates how the Data Quality assessment framework was designed and implemented to meet the specific requirements for evaluating the Data Quality of a plumbline sensor installed at a Portuguese dam.

[2] Github public repository: https://github.com/dioguini/iscte_dsmaster_tese_framework

This framework is not merely an isolated technical too. Its architecture was intentionally designed to align with the principles described in the state of the art, making it suitable for use in professional environments not only for identifying Data Quality issues, but also as a modular tool capable of evolving through the addition of new validation rules or the integration of new instruments.

Each module serves a specific purpose, and the DQ Checks, Threshold Validator, and Visualisation modules are interoperable, meaning they exchange data and operate in a coordinated yet autonomous manner to support execution and result generation.

Also, the Profiling, DQ Checks, and Threshold Validator modules all rely on configurable parameters defined in the manual configuration file, enabling end users to adjust variables and values as needed ensuring better adaptability and execution control of the framework.

Chapter 6

Results Discussion

6.1. Chapter Overview

This chapter presents and analyses the results obtained from the application of the framework developed in this dissertation, using real data from plumb line sensors installed at a Portuguese dam. The main objective is to assess how the framework can identify Data Quality issues by operationalising the defined application layer, while also providing DQ Checks and visualisations that support their critical and contextualised interpretation.

The results analysis is structured according to the layers defined in the framework. It begins with the Management and Policy layer, followed by the modules of the Application Layer and their respective inputs, as defined in the configuration file covering the Profiling, DQ Checks, and Threshold Validator modules, and concluding with the analysis of the visual outputs generated.

Secondly, the implications of these results are discussed in terms of data reliability and their impact on the structural monitoring of the dam.

Finally, an integrated analysis is conducted, relating the results to the expected behaviour of the structure.

This chapter also discusses the contribution of the framework to the state of the art, its advantages and limitations, as well as the potential for future use and evolution.

6.2. Results

This subsection shows and evaluates the outputs provided by the framework, demonstrating its capacities. Here a subset of all the results is presented. It was selected a specific sensor named specifically for this dissertation as “00000” for automatic readings in 2024 and Completeness DQ Dimension. Also, a subset of the Profiling output is presented for 2024 year, since this module is not focused in any DQ Dimension in specific. All the other output results from the Application Layer for the different modules and DQ Dimensions can be consulted in the Appendix E.

The results presented in this dissertation (including Appendix E) are the command line result execution of the following command: *python .\src\dq_framework\dq_framework_exec.py --instrument plumbline --mode all --id_list 00000 --year_list 2024*.

Firstly, the main output directory structure is created under */data/out/* directory as Figure 7 Figure 6 below shows:

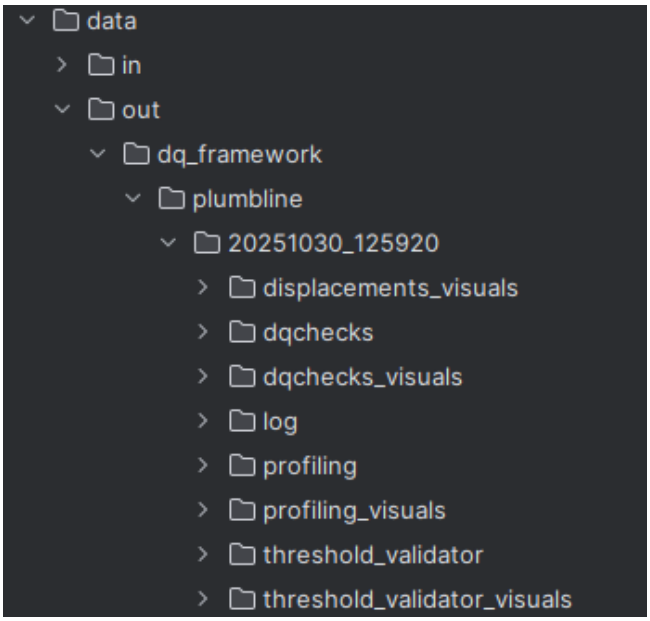


Figure 7. Output directories for the execution

As seen, the different folders are created as the expected framework behaviour. Inside each folder we have the different Application Layer modules outputs. Figure 8 shows the different Profiling module outputs under *profiling/* and *profiling_visuals/*. In the image all the selected files represent the Profilng outputs analysed in this section, for 2024 and automatic readings. All the files are accordingly the output file naming convention defined in the Appendix D.

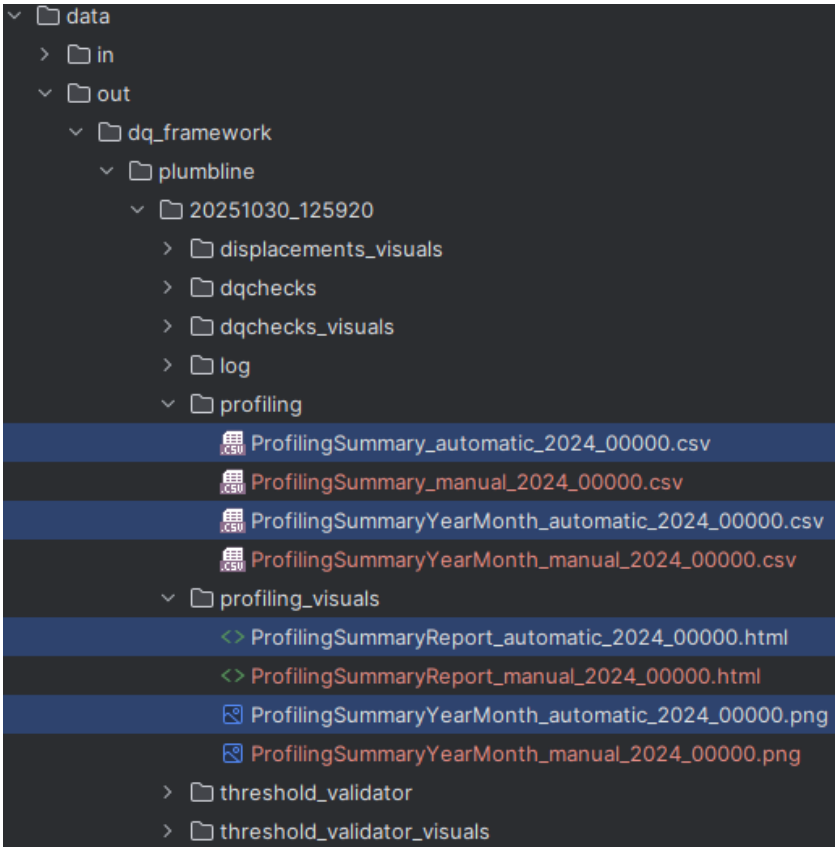


Figure 8. Different Profiling modules outputs

Below, Table 19 shows the different selected profiling columns for automatic readings in *manual.json file – ANO*, *MES*, *DIA*, *HORA*, *MINUTO*, *DESLOCRADIALABS*, *DESLOCTANGABS* as metrics produced by the *Ydata Profiling* library for each one of these columns. By the *min* and *max* value for *ANO* column, it is possible to conclude that the year analysed is 2024.

Table 19. Ydata-Profiling .csv output for automatic reading mode

	ANO	MES	DIA	HORA	MINUTO	DESLOCRADIALABS	DESLOCTANGABS
count	6144.0	6144.0	6144.0	6144.0	6144.0	6144.0	6144.0
mean	2024.0	7.703	15.616	11.447	0.962	-2.477	-2.468
std	0.0	2.817	8.733	7.012	5.444	1.547	0.334
min	2024.0	3.0	1.0	0.0	0.0	-6.30	-4.41
25%	2024.0	5.0	8.0	6.0	0.0	-3.43	-2.61
50%	2024.0	7.0	16.0	12.0	0.0	-2.72	-2.37
75%	2024.0	10.0	23.0	18.0	0.0	-1.31	-2.25
max	2024.0	12.0	31.0	23.0	36.0	1.78	-0.77

Figure 9 shows one of the possible outputs of the *.html* report. In this case it shows the *DIA* variable distribution readings across the year.

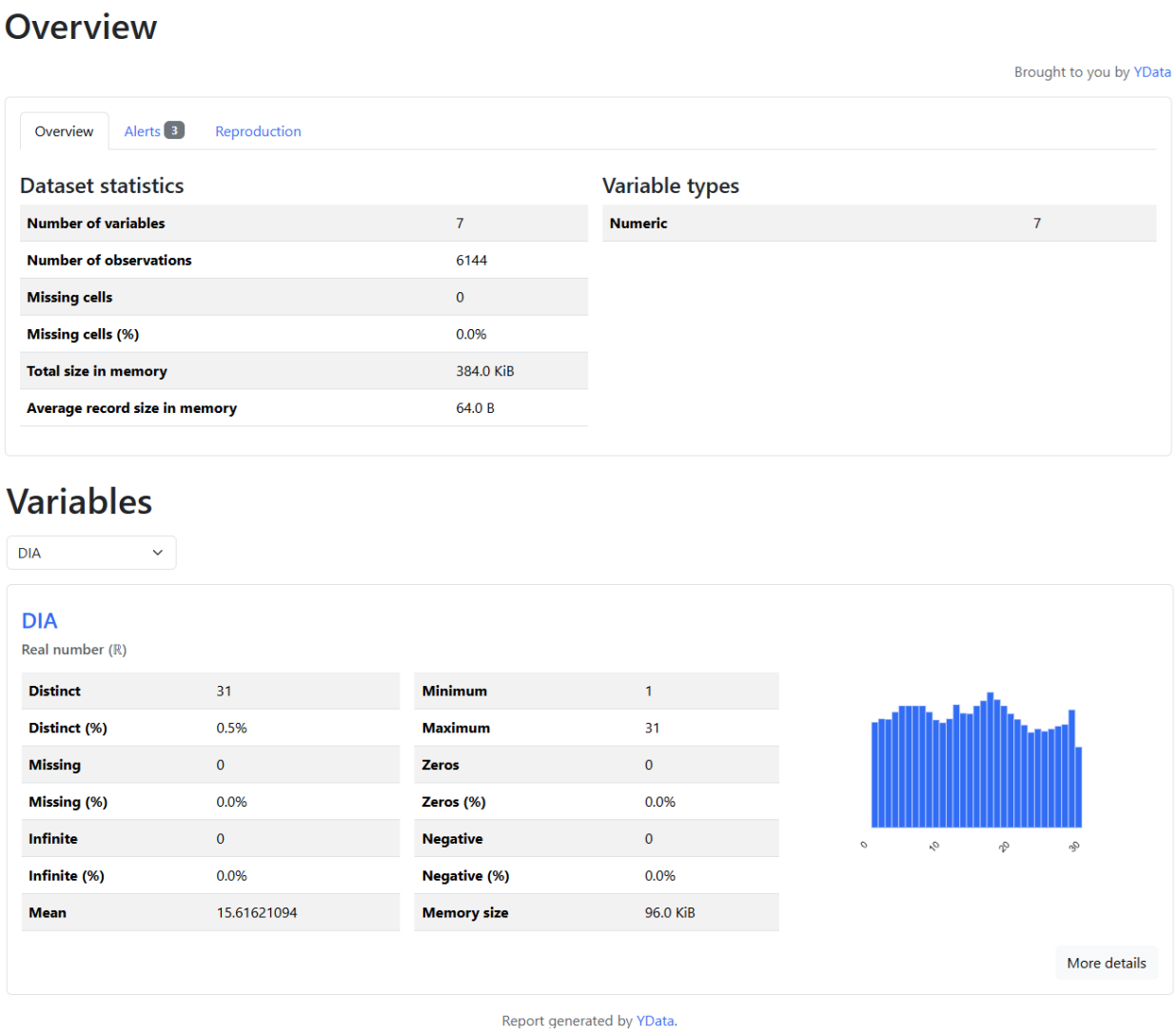


Figure 9. Ydata-Profiling output for DIA variable in automatic read mode (.html)

Table 20 represents the output of the year-month Profiling, showing the readings distribution for the different months for 2024. The output is automatically sorted by *DATA_MONTH* field, representing the month first day of the month.

Table 20. Csv output with year-month distribution values for automatic reading mode – year 2024

PLUMBLINE	READ_MODE	YEAR_MONTH_STR	DATA_MONTH	TOTAL_RECORDS
00000	automatic	2024_March	2024-03-01	298
00000	automatic	2024_April	2024-04-01	719
00000	automatic	2024_May	2024-05-01	635
00000	automatic	2024_June	2024-06-01	720
00000	automatic	2024_July	2024-07-01	739
00000	automatic	2024_August	2024-08-01	650
00000	automatic	2024_September	2024-09-01	334
00000	automatic	2024_October	2024-10-01	593
00000	automatic	2024_November	2024-11-01	716
00000	automatic	2024_December	2024-12-01	740

This table serves as input for the visualisation in below Figure 10. The total number of readings is in the chart's title. This figure shows the monthly total number of readings register by sensor 00000 for 2024.

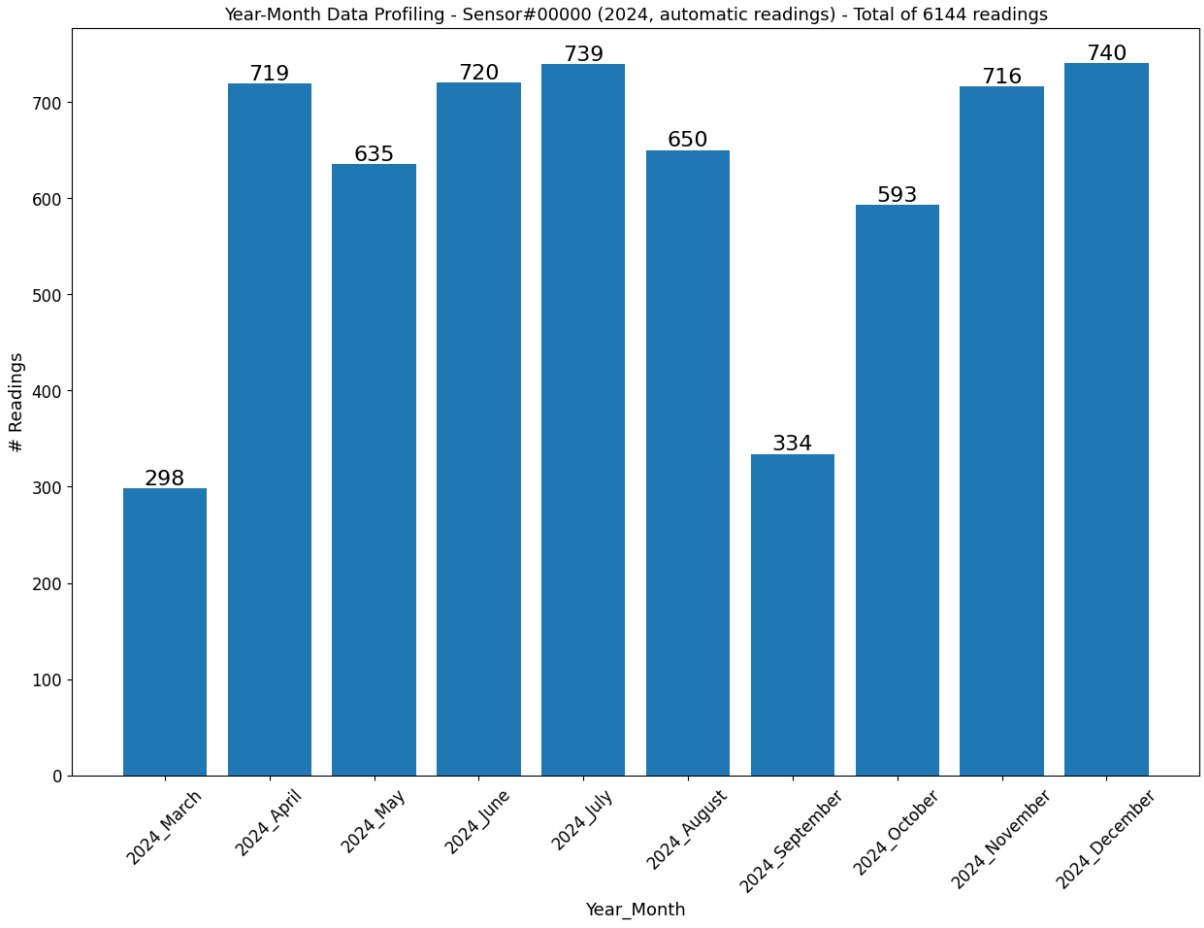


Figure 10. Year-month distribution visualization for automatic reading mode – year 2024

Similarly to Figure 8, below Figure 11 shows the different DQ Checks module outputs *dq_checks/* and the respective Reporting and Visualisation modules output under *dqchecks_visuals/*. In the image all the selected files represent the DQ Checks outputs analysed in this section for 2024 automatic for Completeness DQ Dimension. All files are accordingly the output file naming convention defined in the design chapter, accordingly in Appendix D.

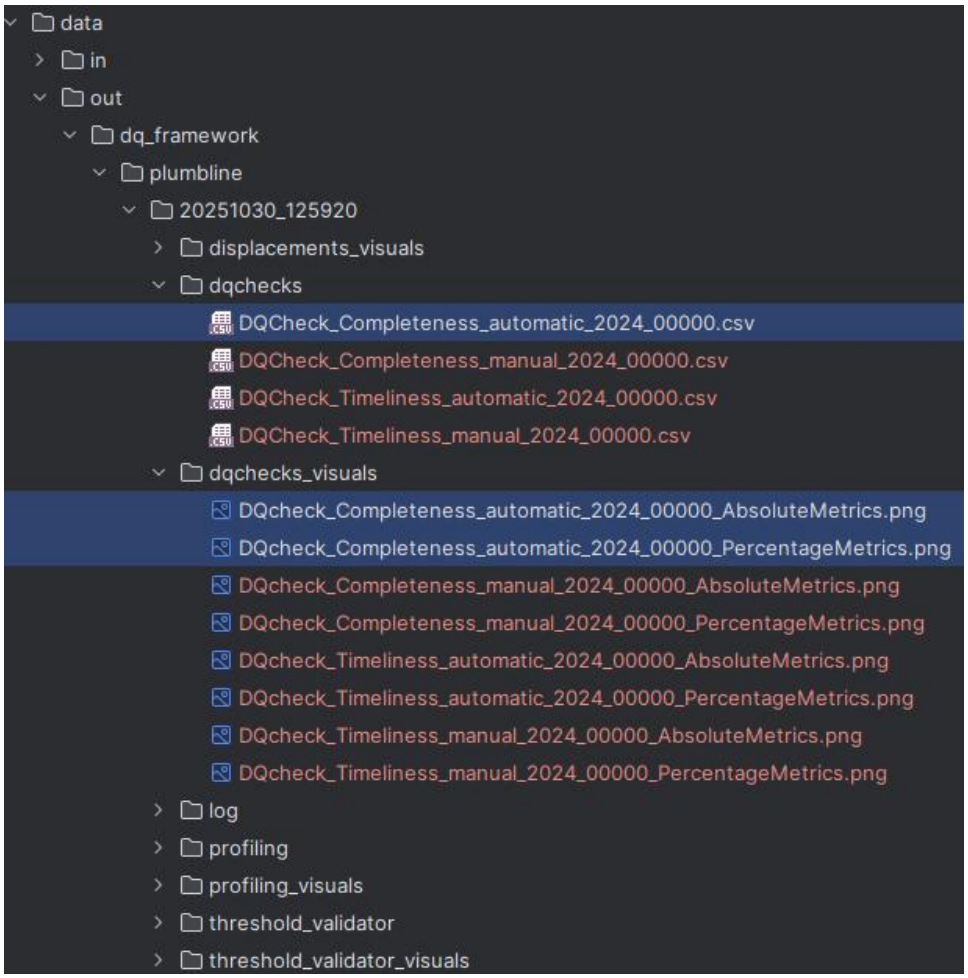


Figure 11. DQ Checks Outputs in output directory

By observing Table 21 and carefully reading results we can conclude that for Completeness DQ Dimension we have two DQ Check *hours_with_multiple_records_abs* and *redundant_records_hours_abs* that have values greater than 0, meaning for the first one, *hours_with_multiple_records_abs* , we have a total of 191 records represents more than one reading per hour, while *redundant_records_hours_abs* DQ Check means that there are a total of hours that have more one or more records per hour.

Table 21. Data Quality Checks .csv output file for automatic read mode – Completeness DQ Dimension

plumbline_id	year	read_mode	dq_dimension	dq_check	value
00000	2024	automatic	Completeness	days_with_multiple_records_abs	0.0
00000	2024	automatic	Completeness	days_with_multiple_records_prctg	0.0
00000	2024	automatic	Completeness	hours_with_multiple_records_abs	191.0
00000	2024	automatic	Completeness	hours_with_multiple_records_prctg	0.03
00000	2024	automatic	Completeness	minutes_with_multiple_records_abs	0.0
00000	2024	automatic	Completeness	minutes_with_multiple_records_prctg	0.0
00000	2024	automatic	Completeness	missing_DESLOCRADIALABS_abs	0.0
00000	2024	automatic	Completeness	missing_DESLOCRADIALABS_prctg	0.0
00000	2024	automatic	Completeness	missing_DESLOCTANGABS_abs	0.0
00000	2024	automatic	Completeness	missing_DESLOCTANGABS_prctg	0.0
00000	2024	automatic	Completeness	missing_data_day_abs	0.0
00000	2024	automatic	Completeness	missing_data_day_prctg	0.0
00000	2024	automatic	Completeness	missing_data_hour_abs	0.0
00000	2024	automatic	Completeness	missing_data_hour_prctg	0.0
00000	2024	automatic	Completeness	missing_data_minute_abs	0.0
00000	2024	automatic	Completeness	missing_data_minute_prctg	0.0
00000	2024	automatic	Completeness	redundant_records_day_abs	0.0
00000	2024	automatic	Completeness	redundant_records_hours_abs	252.0
00000	2024	automatic	Completeness	redundant_records_minutes_abs	0.0

In this step, two outputs are created as mentioned in the implementation chapter allowing to have both percentage and Absolute DQ Check type separated on the respective plots.

At this point it is important to distinguish both DQ Checks: *hours_with_multiple_records_abs* only identifies (counts) hours that have more than the maximum allowed records defined in the configuration file *manual.json,* while *redundant_records_hours_abs* goes beyond and sums all the records for all the hours that are above the maximum allowed records. It is in this step that *Observability* layer becomes crucial for future steps, because it allows the analysts to understand sensor behaviour.

For this case in particular the detailed records were identified for both of the DQ Checks, serving so the Observability layer. Taking one example for each one of the DQ Checks, below Table 22 and Table 23 show the difference between both DQ Checks: for the records ANO=2024, MES=4, DIA in (1,2,4) tables show a different TOTAL value for hour (HORA) 1. Table 22 identifies each one of the records as a *count* function (DQ Check is *hours_with_multiple_records_abs*), while in Table 23 *2* means that hour *1* have two redundant records besides the first one.

Table 22. hours_with_multiple_records_abs sample

ANO	MES	DIA	HORA	TOTAL
2024	3	14	18	1
2024	3	20	15	1
2024	3	21	8	1
2024	3	26	1	1
2024	3	31	0	1
2024	4	1	1	1
2024	4	2	1	1
2024	4	3	1	1
2024	4	4	1	1

Table 23. redudant_records_hours_abs sample

ANO	MES	DIA	HORA	TOTAL
2024	3	14	18	1
2024	3	20	15	1
2024	3	21	8	1
2024	3	26	1	1
2024	3	31	0	1
2024	4	1	1	2
2024	4	2	1	2
2024	4	3	1	1
2024	4	4	1	2

Table 24 below shows a sample for detailed records for each one of the hours identified above.

Table 24. Detailed observations

ANO	MES	DIA	HORA	MINUTO	DESLOCRADIALABS	DESLOCTANGABS
2024	3	14	18	0	-5.04	-2.29
2024	3	14	18	36	-5.09	-2.28
2024	3	20	15	0	-4.53	-2.02
2024	3	20	15	36	-4.30	-2.07
2024	3	21	8	0	-4.48	-2.18
2024	3	21	8	6	-4.44	-2.32
2024	3	26	1	0	-4.40	-2.07
2024	3	26	1	36	-4.24	-2.15
2024	3	31	0	0	-5.07	-2.03
2024	3	31	0	36	-5.06	-2.17
2024	4	1	1	0	-4.91	-2.33
2024	4	1	1	6	-5.11	-2.37

2024	4	1	1	36	-4.94	-2.24
2024	4	2	1	0	-5.00	-2.30
2024	4	2	1	6	-4.98	-2.30
2024	4	2	1	36	-5.01	-2.24
2024	4	3	1	0	-5.11	-2.42
2024	4	3	1	6	-5.13	-2.38
2024	4	4	1	0	-5.16	-2.25
2024	4	4	1	6	-5.17	-2.32
2024	4	4	1	36	-5.22	-2.21

All the outputs for each one of the DQ Checks for both Completeness and Timeliness DQ Dimensions, can be checked in the Appendix E.

Also, for *hours_with_multiple_records _abs* there is the *hours_with_multiple_records_prctg* DQ Check, with a value of 0.03, meaning that 3% of the records have at least one duplicated value (this is the result of 191 records in a total of 6144).

For Timeliness DQ Dimension the results are presented in table below, where it is possible to check the results (*value* column) for each one of the DQ Checks executed (*metric* column). In the table, *plumbline_id* column represents the plumbline analysed, *year* column for the year data, for automatic reading mode (*read_mode* column) for Timeliness DQ Dimension (*dq_dimension* column).

Table 25. Data Quality Checks .csv output file for automatic read mode – Timeliness DQ Dimension

plumbline_id	year	read_mode	dq_dimension	dq_check	value
00000	2024	automatic	Timeliness	data_day_density_abs	20.9
00000	2024	automatic	Timeliness	data_hour_density_abs	0.87
00000	2024	automatic	Timeliness	data_minute_density_abs	0.01
00000	2024	automatic	Timeliness	days_apart_previous_record_count_abs	268
00000	2024	automatic	Timeliness	days_apart_previous_record_proportion_prctg	0.04
00000	2024	automatic	Timeliness	hours_apart_previous_record_count_abs	221
00000	2024	automatic	Timeliness	hours_apart_previous_record_proportion_prctg	0.04
00000	2024	automatic	Timeliness	interval_days_abs	0.28
00000	2024	automatic	Timeliness	interval_days_prctg	0.05
00000	2024	automatic	Timeliness	interval_hours_abs	4.63
00000	2024	automatic	Timeliness	interval_hours_prctg	1.15
00000	2024	automatic	Timeliness	interval_minutes_abs	278.04
00000	2024	automatic	Timeliness	interval_minutes_prctg	68.81
00000	2024	automatic	Timeliness	max_gap_days_abs	13
00000	2024	automatic	Timeliness	max_gap_hours_abs	308
00000	2024	automatic	Timeliness	max_gap_minutes_abs	18480

00000	2024	automatic	Timeliness	minutes_apart_previous_record_count_abs	221
00000	2024	automatic	Timeliness	minutes_apart_previous_record_proportion_prctg	0

All the files are accordingly the output file naming convention defined in the design chapter, accordingly in the Appendix D.

Threshold Validator module is designed as DQ Checks module: first it gets input – the *dqchecks* output – and then compares each DQ Check with the defined thresholds in *manual.json file*. The outputs are then generated in the Reporting and Visualization Layer.

Below, Figure 12 shows the different Threshold Validator module outputs under *dq_checks/* and Reporting and Visualization Layer under *dqchecks_visuals/*. In the image all the selected files represent the Threshold Validator outputs analysed in this section for 2024 automatic and Completeness DQ Dimension directory.

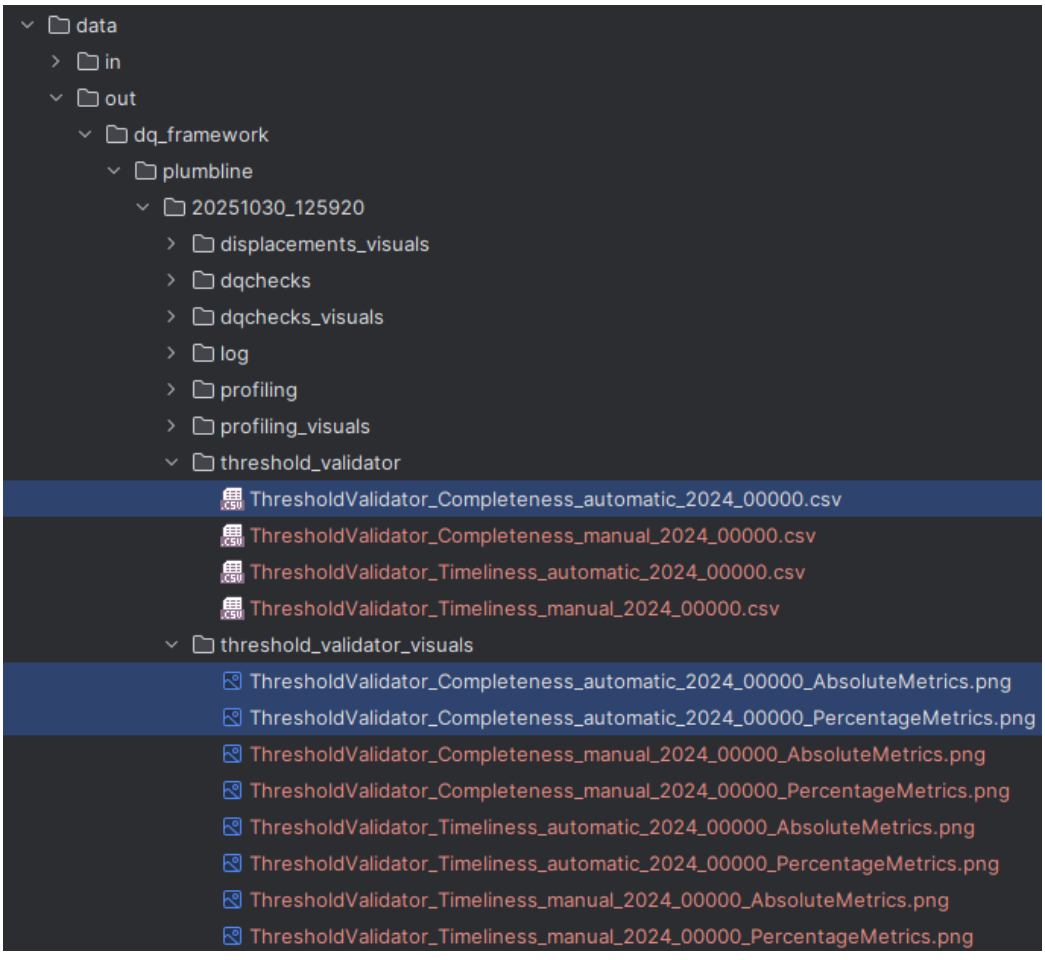
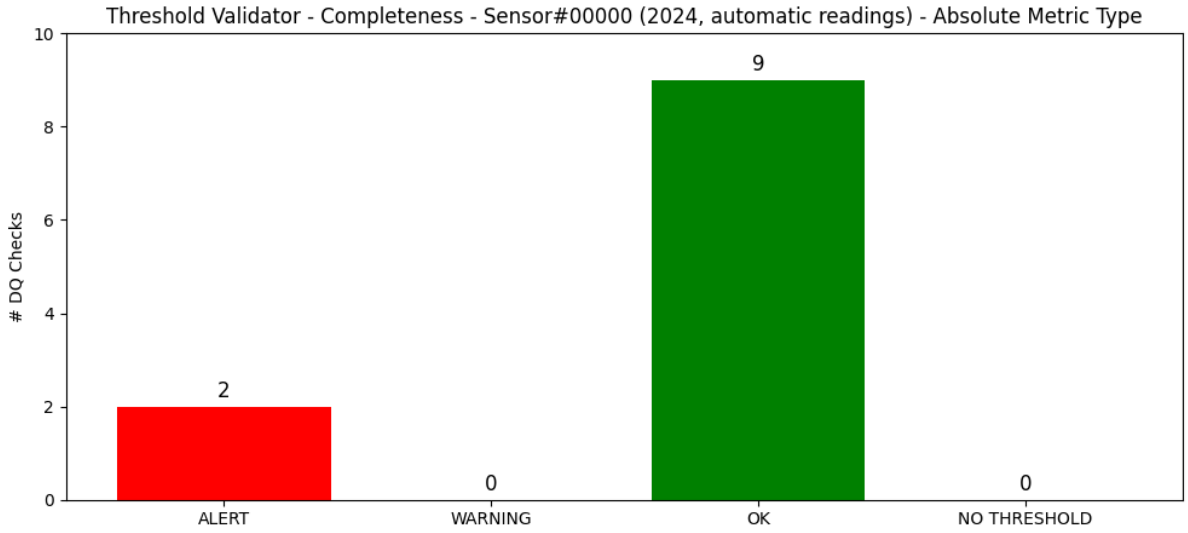


Figure 12. Threshold Validator outputs

In Appendix D, also represents the *.csv* output file with all the threshold values calculated in this module, in *threshold_validator.py* script. The output has the column *status* showing the defined Alert type based on the *value* output calculated column, that represents the threshold value calculated. The *value* column is compared with *threshold* column (value defined in *manual.json* file) and based on the limit factor calculation it raises the Alert type (*status* column). Additional columns like *threshold_diff* and *threshold_deviation_pct* are in the file. The first one, *threshold_diff*, represents the absolute difference between threshold value allowed and the value calculated, while *threshold_deviation_pct* represents the percentage deviation between *threshold* and calculated threshold value column *value*. This value becomes 100 (100%) when the defined threshold is 0 and the calculated DQ Check value is non-zero.

Below, Figure 13 represents the Alert distribution by Alert type. Different visualizations are created accordingly the DQ Check type, allowing the different DQ Check type (Absolute and Percentage) to be analysed in their specific report.



Metric	Metric Value	Threshold Value	Threshold Difference	Threshold Deviation (Percentage)	Alert Label
hours_with_multiple_records_abs	191.0	0	191.0	100.0	ALERT
redundant_records_hours_abs	252.0	0	252.0	100.0	ALERT
days_with_multiple_records_abs	0.0	0	0.0	0	OK
minutes_with_multiple_records_abs	0.0	0	0.0	0	OK
missing_data_day_abs	0.0	0	0.0	0	OK
missing_data_hour_abs	0.0	0	0.0	0	OK
missing_data_minute_abs	0.0	0	0.0	0	OK
missing_DESLOCRADIALABS_abs	0.0	0	0.0	0	OK
missing_DESLOCTANGABS_abs	0.0	0	0.0	0	OK
redundant_records_day_abs	0.0	0	0.0	0	OK
redundant_records_minutes_abs	0.0	0	0.0	0	OK

Figure 13. Threshold Validator Reporting output for automatic reading mode – Completeness DQ Dimension (Absolute DQ Check type)

The Report and Visualization Layer also generates visualisations to analyse *DESLOGRADIALABS* and *DESLOCTANGABS* values, represented respectively in figures Monthly Average Radial Displacement and Monthly Average Tangential Displacement. Both visualisations are under *displacements_visuals* in the output directory as Figure 14 and both are accordingly file naming convention in the Appendix D. Figures Figure 15 and Figure 16 represent the selected files in Figure 14 and are respectively Monthly Average Radial Displacement and Monthly Average Tangential Displacement. For both, in the same chart are represented in a violin plot the distribution of the monthly values for each of the variables, combined with the blue line representing the average value for each values: Figure 15 uses *DESLOGRADIALABS* column data and Figure 16 uses *DESLOCTANGABS* column data.

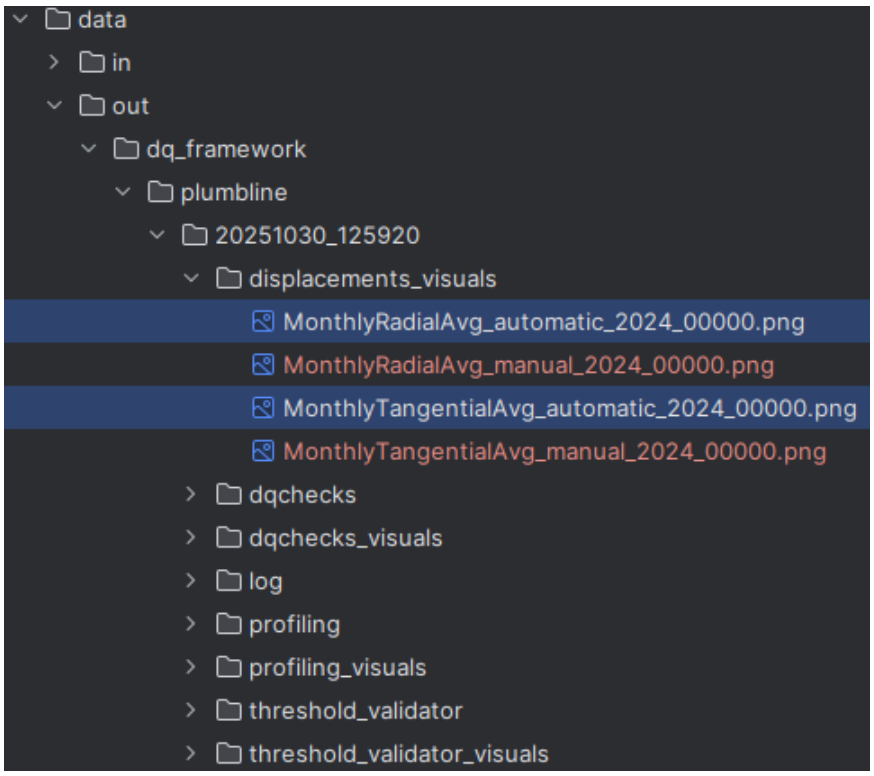


Figure 14. Displacement visuals

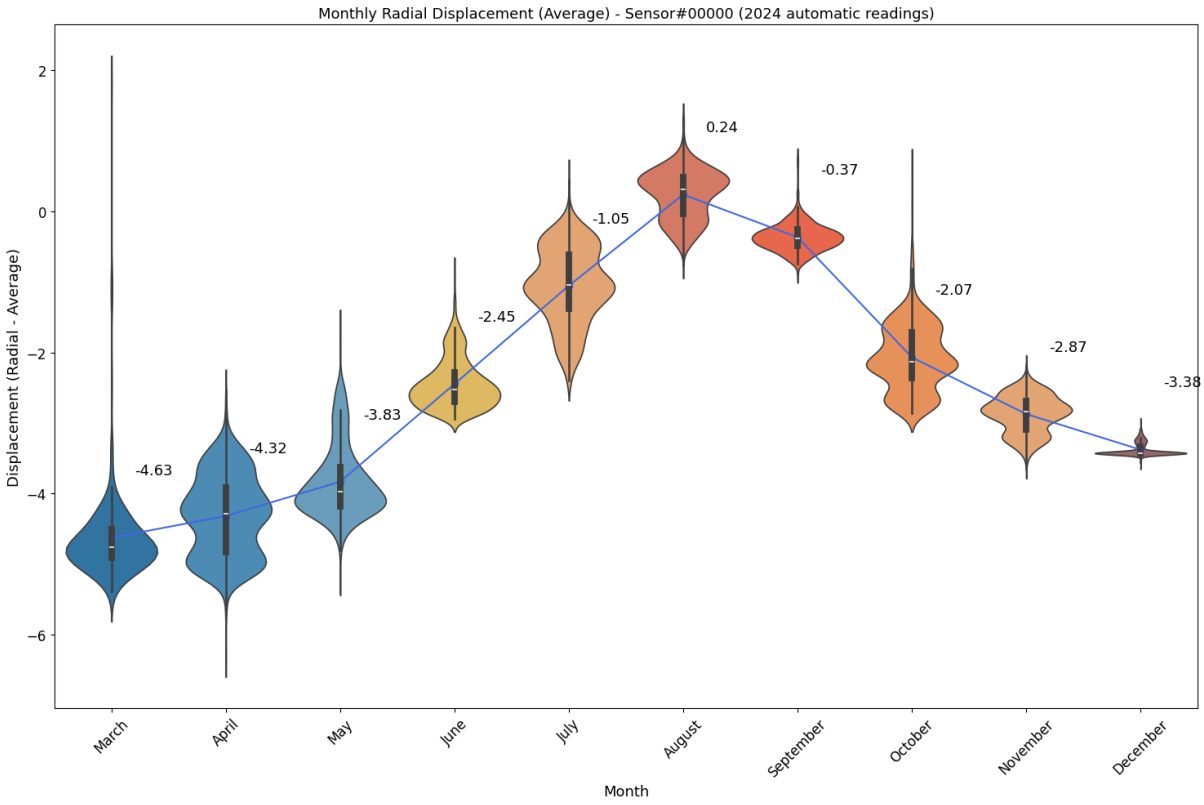


Figure 15. Monthly Average Radial Displacement for automatic reading mode

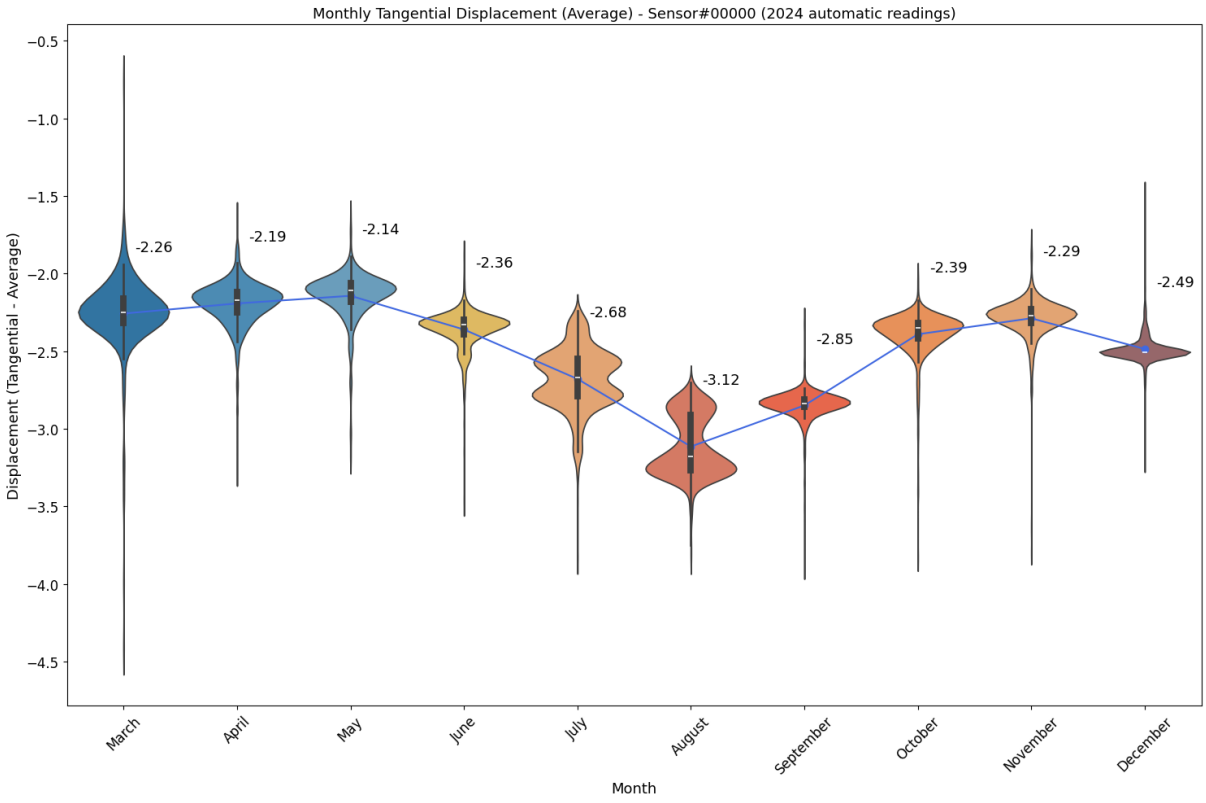


Figure 16. Monthly Average Tangential Displacement for automatic reading mode

Lastly the Feedback Layer has all the outputs generated for the DQ Checks mentioned before:

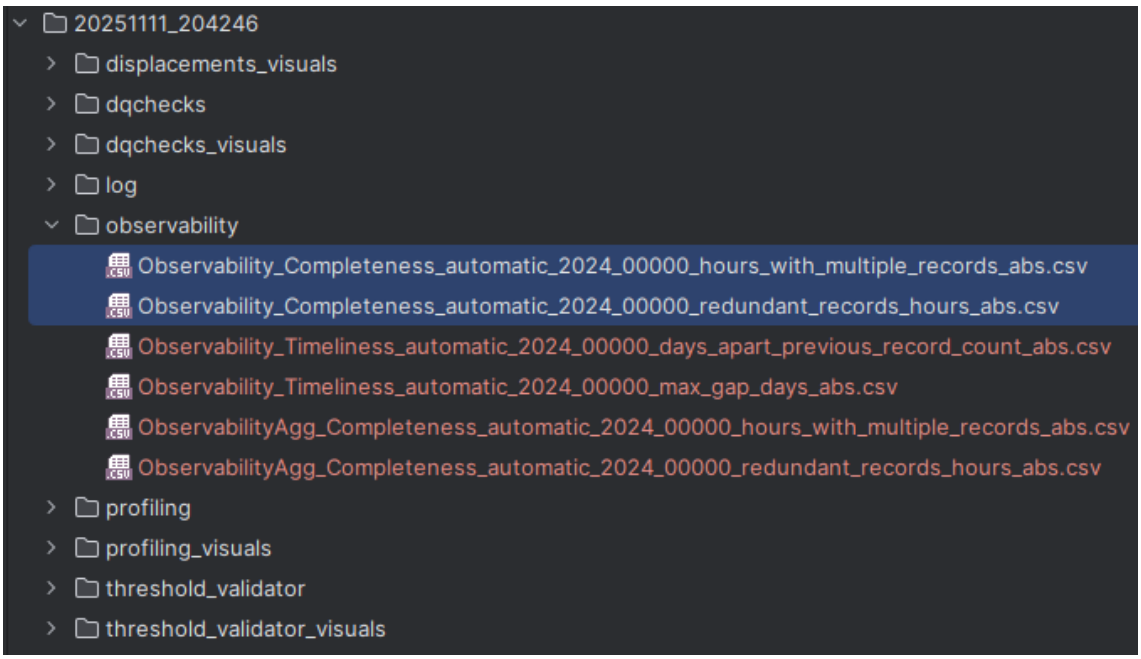


Figure 17. Feedback layer files output

6.3. Results Discussion

This section discusses the results obtained directly from the framework focusing on assessing Data Quality for a specific the specific plumb line sensor 00000 for 2024 data. In this section all the outputs from previous section (Completeness DQ Dimension, automatic reding mode) are taking into consideration, along with the outputs in the Appendix C that focus on Timeliness DQ Dimension automatic and manual readings, and also Completeness DQ Dimension manual readings.

6.3.1. Profiling

The profiling layer enabled a quick standardisation analysis of the input data files, allowing for an initial understanding of whether any data gaps were present.

For automatic reading mode, *count* metric value for all the variables, in Table 19 indicates that the file used does not have any gaps, and has a total of 6144 records. By observing the *min*, *max*, *std*, and *mean* values for *HORA* (hour) and *MINUTO* (minute) it is possible to conclude that the readings are not being taken always at the same time and analysing the same metrics but now for *DESLOCRADIALABS* and *DESLOCTANGABS* it is possible to conclude that the different dam movements (Radial and Tangential) are being registered.

The total number of readings, 6144, is accordingly with the first profiling analysis as seen in Table 19 for the *count* metric and in Figure 10. Also, in Figure 10 it can be observed that the total number of readings for sensor 00000 in automatic mode averages close to 700 readings per month. This aligns with the expected acquisition frequency (hourly), considering that a typical month has approximately 730 hours. However, several issues are noticeable: the year 2024 is incomplete: the sensor shows no data at all for January and February. Also, for the months of March and September, both contain only about half of the expected readings.

In the Appendix D it is clear that for manual reading mode, the number of readings is consistently two per month, which aligns with the initial expectations defined for manual mode. All months are represented, but March contains an outlier with three total readings. Furthermore, it is evident that two months contain only one reading, similarly to the automatic mode, though these months differ specifically, February and September.

6.3.2. Completeness DQ Dimension

The framework, by automatically identifying the presence of missing records and classifying them according to the configured rules and thresholds, enables the objective quantification of data completeness and provides early warning of critical reading failures, particularly for automatic readings obtained via sensors. The absence of data can compromise the ability to detect real anomalies in the structural displacements of the dam, leading to incomplete or biased analyses.

In relation to the data analyzed and the results obtained in the context of this dissertation, the framework was able to identify duplicated and redundant records for the automatic reading mode as mentioned in Table 22 and Table 23. From this analysis, it is possible to observe 191 duplicated records for certain hours although they represent only 3% of total records, as well as 252 redundant records. These two DQ Check are complementary: through the analysis of *redundant_records_hours_abs*, it is possible to conclude that the number of redundant records per hour is greater than one, indicating readings that exceed the expected cadence (1 reading/hour). This reinforces the presence of redundancy, showing that in 191 distinct hours, there were multiple records.

Both results suggest that the sensor is recording more frequently than expected per hour.

One possible explanation may be that the threshold value is too conservative, or it could indicate configuration errors or noise in the data collection process. However, considering that only 191 distinct records are duplicated (representing 3% of the dataset), the most likely cause lies with the sensor itself.

6.3.3. Timeliness DQ Dimension

The evaluation of the Timeliness dimension allows us to understand the regularity and continuity of the sensor readings over time.

From the analysis of the automatic mode, as shown in Table 25**Error! Reference source not found.**, it can be observed that the daily and hourly density is close to the threshold values defined in the configuration file (*data_day_density_abs* = 20.09, *data_hour_density_abs* = 0.87, close to 1), showing that the sensor is consistent with the defined configuration meaning that the sensor is registering 20 readings per day and 0.87 per hour.

This sensor also shows a high frequency of readings on a daily basis (*interval_days_abs* = 0.28), but also *interval_hours_abs* = 4.63 and *interval_minutes_abs* = 278.04, suggesting that the average acquisition intervals are relatively short. However, despite these values, the sensor presents failures such as significant temporal gaps, with the largest interval between consecutive records being 13 days (*max_gap_days_abs* = 13.0), which represents a critical failure in data collection. This exact value conversion corresponds to 312 hours. However in Appendix E output shows a total of 18,516 minutes (308 hours) showing that the interval wasn't perfect, concluding the sensor started recording readings in this timeframe. These values also show that in terms of hourly and minute-level granularity, the largest observed gaps were 308 hours (18,516 minutes), respectively, confirming a consistent pattern of data loss across multiple periods. Considering that the DQ Checks are based on chronologically ordered data down to the minute, we can confidently state that the missing data identified in the initial profiling, as seen in Figure 10 is directly related to the absence of data confirmed in this analysis, however it's not possible to understand if the absence of records is at the beginning or at the end of the month.

The sensor also shows a high number of interruptions that exceed the maximum thresholds defined for the variables: *max_interval_days* (0), *max_interval_hours* (1), and *max_interval_minutes* (60). There is a direct relationship between these variables and the corresponding analysis results, as they are all based on the timestamp of observations which are chronologically ordered. As a result, the DQ Checks *days_apart_previous_record_count_abs*, *hours_apart_previous_record_count_abs*, and *minutes_apart_previous_record_count_abs* display values of comparable magnitude.

Additionally, 268 instances of time gaps greater than one day, and 221 instances greater than one hour, were recorded, revealing a highly irregular distribution of data over time. All these alerts are clearly presented in Threshold Validator visualization output for automatic reading mode – Timeliness DQ in Appendix E.

In contrast, manual readings present three alerts, notably a severe punctuality failure, with a maximum gap of 39 days (*max_gap_days_abs*). Result can be checked in the Appendix E.

These types of temporal quality issues can significantly impact the interpretation of the dam's structural displacements. Although not the specific case for this sensor, in the event of missing data during months typically associated with heavy rainfall, this behavior could delay or hinder the identification of seasonal deformation patterns, such as those driven by reservoir level variations.

Moreover, the loss of temporal granularity (gaps in hourly or daily data continuity) may also mask short-term anomalies, preventing early detection of potential structural problems. The framework proved essential in identifying these limitations, offering objective indicators of delays or failures in data acquisition, and thus promoting a proactive management of the timeliness dimension in sensor Data Quality. All the alerts are presented in the Appendix E.

6.3.4. Monthly Radial and Radial displacements

The outputs generated for 2024 year for automatic readings in Figure 15 and Figure 16 are useful to understand the expected tangential and radial movements of the dam, as well as for quick identification of outliers relative to the average values.

It is easily noticeable from the figures that the values fluctuate consistently within the same range for both automatic and manual readings. However, due to the difference in reading frequency, manual readings appear more standardised, while sensor data shows greater oscillations, even though both use two decimal places. Still, it is evident that the data allows for the interpretation of the dam's expected behaviour. Typically, months associated with heavier rainfall cause the dam to show negative radial movement (forward, towards the downstream direction), while warmer months show positive values (indicating a slight "retraction" of the structure). The same seasonal pattern is observed in the tangential movement, for both types of readings.

The results show that the radial and tangential displacement values obtained from both acquisition modes are consistent with the expected seasonal movement of the dam and are also consistent between each other.

It is also worth mentioning that manual mode alone may be sufficient for analysis, since it typically provides two readings per month. However, as previously seen, there are months with only one reading, which results in a single value being displayed on the graph, making it impossible to observe oscillations. Outputs are the Appendix E helping visualising all the info. On the other hand, automatic mode better captures the dam's behaviour, due to its higher frequency of readings. Even in months with approximately half the expected number of readings, both radial and tangential movements are still represented.

It is important to highlight that no quality assessment has yet been performed for concordance or accuracy of these acquisition modes. This should be considered a future task to enrich this research.

6.3.5. Feedback Layer outputs analysis

The feedback layer adds significant value to the entire data analysis cycle of the framework, as it allows the observation of all results produced by the evaluated DQ Checks that generated outputs.

Within the scope of this dissertation, it is possible to conclude that, for the automatic acquisition mode, sensor 00000 occasionally registers readings beyond the expected frequency. This conclusion relies on the results of the DQ Checks *hours_with_multiple_records_abs* and *redundant_records_hours_abs*. When examining the generated outputs, it becomes evident that the implemented DQ Checks follow the defined logic, producing an output that correctly identifies, for the Feedback layer, all records flagged by both checks.

Considering the data lifecycle presented in Figure 2 the Observability layer becomes particularly relevant, as it provides detailed results to any analyst, enabling the identification of failures concerning a specific sensor.

6.3.6. Manual vs Automatic acquisition

The framework developed in this dissertation enabled a comparative analysis of the two main data acquisition modes, manual and automatic, highlighting significant differences in the type and frequency of Data Quality issues associated with each.

In the manual mode, data is often entered by human operators, which increases the risk of human error, such as omissions in data collection, incorrect value entry, or scheduling failures. This mode tends to present larger temporal gaps and irregular data cadence, making it more vulnerable to the loss of temporal granularity, which is essential for accurate structural monitoring.

In contrast, automatic acquisition benefits from regular and systematic collection but introduces other challenges. Automated sensors are susceptible to technical faults, as described by (Teh et al., 2020), including outliers, biases, deviations, missing values, and uncertainties, which must be detected, quantified, and corrected to improve the quality of sensory data. Similarly, (Liu et al., 2020) report issues such as measurement errors, noise, artifact errors, data frame distortion, dirty data, missing updates, data loss, and delayed data transmission.

In the specific case of sensor 00000, used in this study, there were no missing records for the expected fields, confirming the completeness of the available dataset.

The framework also revealed that for both acquisition modes, there are two months per year where the number of readings is approximately half of what is expected, which may indicate intermittent acquisition failures.

While both manual and automatic modes allow for the identification of the dam's expected seasonal behaviour, the automatic mode demonstrates greater sensitivity, including what appear to be outliers (outside the scope of this dissertation).

Therefore, the framework proves essential for monitoring both types of data acquisition, manual and automatic, and reinforces that the choice between these modes should be accompanied by robust Data Quality verification mechanisms, as provided by this work, to ensure the reliability and trustworthiness of the collected data.

6.4. Chapter Summary

This framework aligns with the existing literature not only because it is based on ISO standards and DAMA principles, but also because it is framed and consistent with previous research presented in the literature review chapter, including the identification of common data issues highlighted therein.

The results discussion reveals that, for the Completeness dimension, there is 100% overall coverage of the records, indicating no missing data in either the sensor-based readings or the manually collected records. However, since the data is time-series, the Timeliness dimension became essential to fully understand the data context, exposing issues in both sensor acquisition and manual reading processes. In general, the sensor presents an intermittent behaviour: although the average record density suggests good performance, the presence of multiple temporal gaps and intervals of up to 13 days without records compromise the timeliness of the information. These findings are supported by the profiling outputs, which show a continuous data absence rather than an intermittent one.

On the other hand, manual readings are sporadic but fall within the acceptable time limits, with few outliers.

Both types of readings reflect the expected seasonal behaviour of the dam, as observed through average radial and tangential displacement values. However, no evaluation was performed on the accuracy or consistency of these values, leaving room for future investigations in this area.

These behaviours are particularly critical in continuous monitoring contexts, where missing data can hide significant events or compromise the structural interpretation of the dam’s behaviour.

CHAPTER 7

Conclusion

7.1. Summary of Results

The development of the framework followed a methodological process, aligning quality objectives with operational requirements and establishing a robust foundation for quality management throughout the data lifecycle in sensor networks. It is important to emphasise that the definition of a Data Quality Framework should not be limited to the application of technical checks alone but rather be understood as a multidimensional structure that integrates people, processes, metrics, and technology. Its central purpose is to ensure that the collected data is not only relevant, but above all reliable, supporting informed decision-making in critical contexts. This integrated vision forms the cornerstone of the framework proposed in this dissertation, as demonstrated in the following sections.

The work carried out in this dissertation aimed to design and implement a framework for assessing the quality of data from sensors, with its practical implementation carried out in Python, using real data from a plumbline sensor installed at a Portuguese Dam. With a view to integration into a structural monitoring context for dams, the framework focuses essentially on covering all necessary steps to manage the data lifecycle, from identifying data sources to the final detailed feedback phase.

Based on a systematic literature review, two priority dimensions were identified for application in this domain: Timeliness and Completeness. The framework was designed so that the necessary steps allow for a practical, automated, flexible, and interpretable evaluation of these dimensions, integrating functionalities such as:

- The initial definition of a structured template specifying what the framework should consider in practice (e.g., identification of Data Stewards, sensor specifications);
- The generation of data profiling reports based on a predefined list of columns from the manual configuration file.
- The creation of .csv reports for DQ Checks and Threshold Validator.
- The production of interpretive visualisations that support decision-making and data correction if needed.

The framework also supports external configuration and command-line execution, with all outputs generated automatically using a timestamp-based versioning system.

The practical application of the framework using real-world data from plumbline sensors confirmed its capacity to identify data responsibilities and implementation requirements, detect missing values, delays, and gaps in data acquisition. Moreover, it allows for the visual representation of identified issues through graphs, enabling further analysis that can support understanding of the physical behaviour of the structure.

The steps defined in the framework have proven effective as a support tool for monitoring Data Quality in a critical environment, with potential for broader application to other sensors installed in the dam.

7.2. Conclusions Regarding the Research Questions

The research questions outlined at the beginning of this dissertation guided both the conceptual development and practical implementation of the DQ assessment framework in SHM contexts using senor data.

Regarding *RQ1 – What are the Data Quality problems that affect sensors in civil engineering structures?* the analysed data show that sensors installed in civil engineering structure, such as the plumbline studied in this dissertation are subject to multiple Data Quality issues. The most relevant includes prolonged temporal gaps, record redundancy, readings outside the expected cadence, and inconsistencies in temporal granularity. In the specific case of the sensor analysed, interruptions longer than 13 days were identified, as well as multiple instances of duplicated records, exposing limitations in both automatic and manual acquisition modes.

Regarding *RQ2 – What are the relevant Data Quality Dimensions for sensor data in civil engineering structures?* this dissertation focused on two key dimensions: Completeness and Timeliness. The Completeness dimension evaluates whether the expected data are present, covering mandatory fields and the absence of empty or missing records. The Timeliness dimension assesses the temporal adequacy of the data whether they are recorded with the defined cadence, whether there are interruptions or delays, and whether the data maintain consistent granularity over time. Both dimensions proved to be critical to ensure the reliability of monitoring and to enable a correct interpretation of the physical behaviour of the structure.

Finally, for RQ3 – *What are the most common scenarios for this type of problem?* the proposed framework addresses this question by implementing an observability layer allowing data analysts to understand all the outputs Using the observability layers, data specialists can analyse the outputs and identify the possible issues and respective solutions.

7.3. Contributions to Research and Practice

This work contributes to professional practice by providing a concrete solution that improves the governance of sensor data in critical contexts, like exemplified in this dissertation. It reduces the manual effort required for data validation and verification, which is prone to error, by generating standardised reports, with the potential for results to be integrated into other applications.

In terms of research contributions, this work offers an applied approach to Data Quality in time series data from sensor networks based on scalable model capable of supporting the integration of new instruments represented by sensors. The framework provides a foundation for future developments in adaptive and intelligent Data Quality assessment frameworks since it was designed with a clear alignment with consolidated methodologies for implementing Data Quality frameworks, particularly with the principles established by DAMA and the processes defined by the ISO 8000-61 standard. Firstly, the framework follows the four DAMA principles for Data Quality management (DAMA International, 2017a):

1. Governed approach to ensure data is fit for purpose – the framework was designed based on a governed approach, supported by the Governance and Policy Layer, which defines the rules, roles, and responsibilities for managing sensory data.
2. Definition of standards, requirements, and specifications for DQ controls – through the formalisation of requirements, metrics, and thresholds in a dedicated document, ensuring that quality controls are aligned with the data lifecycle.
3. Implementation of processes to measure, monitor, and report DQ levels – operationalised in the Application Layer, where DQ checks are executed, compared against thresholds, and results are reported in a structured way.
4. Identification and promotion of opportunities for DQ improvement – the framework enables incremental evolution and continuous improvement through iterative refinement based on the results obtained.

In addition, the framework also aligns with the principles defined in the ISO 8000-61 standard described by (King & Schwarzenbach, 2020).

Within the scope of the proposed framework, the following process components of ISO 8000-61 are incorporated:

1. Data Quality Planning: The document defined in the Governance and Policy Layer, establishes Data Quality requirements, roles and responsibilities, data flow, Implementation strategy based on available resources and technology.
2. Data Quality Control: The Application Layer ensures the implementation of previously defined data requirements. Also, specifications and work instructions (e.g., file types,

columns to be evaluated, acquisition frequency) are formalised and consistently followed.

3. Data Quality Assurance: the framework reviews Data Quality issues based on generated outputs (e.g., gaps, redundancies, invalid values) and defines assessment metrics and measures the performance of the Data Quality system. Results are then evaluated and analysed to understand the impact of poor Data Quality and to define corrective actions. This step can be performed based on the outputs from both the Visualisation and Reporting Layer and Feedback Layer.

The framework developed aligns with international standards and supports a mature approach to sensor Data Quality management. It considers sensor network characteristics, allowing multiple sensors and acquisition modes (manual and automatic) to be assessed simultaneously through spatially aware DQ Checks applied to time-series data. At the organisational level, these features translate into significant strategic value, allowing for a systematic and auditable record of sensor Data Quality over time. This tool provides a foundation for more informed operational decisions in the context of SHM monitoring. The ability to generate standardised and interpretable outputs for different user profiles (from technical analysts to asset managers) enhances its practical utility and strengthens its role in supporting the governance of sensor data, contributing to greater transparency and traceability.

Additionally, the modular and configurable design of the framework makes it reusable and scalable, allowing it to be applied to new sensors, acquisition modes.

7.4. Critical Analysis

The framework developed in this dissertation provides a contribute at the intersection of Data Quality and structural monitoring, by integrating, in an automated and modular way, two critical dimensions for assessing sensor Data Quality: Timeliness and Completeness.

Unlike generic Data Quality approaches typically applied to traditional information systems based on transactional data, this framework specifically uses time-series sensor data originating from physical environments, such as the real dam data used on this dissertation, and operates at the data acquisition stage, where errors can compromise the entire monitoring and analysis chain

The practical application of the framework has demonstrated a set of significant advantages for the identification of Data Quality issues, such as the automation of the evaluation process, thereby reducing the manual verification workload for engineers and analysts. The presence of a Management and Policy document that spans the entire framework is of critical importance, as it summarises key information for understanding the applied framework identifying data sources, acquisition frequency, Data Stewards, and application requirements such as the DQ Checks to be implemented and the thresholds to be used.

In the Application Layer, the inclusion of a profiling module enables quick descriptive insights into the data for any analyst, either through browser-based reports or through annual and monthly record distribution summaries, generated individually for each sensor. The modular structure of the source code allows for easy extension to new sensors, following a straightforward logic of addition and command-line execution, without needing to modify the existing development logic.

Another key strength of the framework is its flexible execution mode, allowing for centralised parameterisation directly within the execution command enabling combinations of different sensors, reading modes, and specific years for analysis.

The visual reporting layer provides a set of automatically generated visualisations, which greatly support the visual interpretation of anomalies, technical reporting, and understanding of results.

Overall, the framework is standardised with clearly defined naming conventions for variables, functions, and outputs, ensuring ease of understanding not only of the codebase but also of the results. Moreover, the structured and timestamped folder organisation guarantees full traceability of executions, allowing any analyst (even without prior knowledge of the framework) to interpret and navigate through the outputs.

These features make the framework highly suitable for integration into operational pipelines of a SHM environments monitoring system, and potentially applicable to other sensor network contexts.

The modular structure and the separation between validation logic, visualisation, and configuration make the framework highly reusable and extensible, allowing the integration of other structural sensors by simply adjusting sampling parameters and dimension-specific DQ Check.

Another potential extension is the integration with external visualisation or alerting systems, such as Power BI or Grafana, since all output files in .csv format within each module follow a consistent schema. This consistency enables the import of all outputs into a centralised reporting or alerting environment, which can generate richer insights when compared to the basic visualisations presented in this dissertation.

7.5. Limitations Identified

Although the results may be considered a step forward, it is important to acknowledge some limitations of the developed solution. Firstly, the framework does not incorporate machine learning techniques for adaptive anomaly detection, relying instead on fixed or manually validated thresholds. This dependence introduces potential sources of error during execution, either due to incorrect parameter input, accidental file modification or corruption, or the need for a senior analyst with expertise to define values such as thresholds or the *limit_factor*.

Another limitation lies in the fact that the framework does not include a specific module for automatic data correction or imputation. However, this framework gives an important support for these procedures, since on the feedback layer, all the original records that are analysed during the DQ Checks are saved and so, available for analysts and engineers helping on the correction data procedures.

The application was tested only with plumbline sensors, and its extension to other sensor types (especially those with different frequency or dynamic characteristics) may require significant adaptation, including the addition of a new instrument in the manual configuration file and the definition of new Data Quality checks and threshold validators.

Nevertheless, these limitations open the door to future developments and enhancements, as discussed in the following subchapter.

7.6. Future Work

Despite the progress achieved, this work opens several opportunities for future developments.

By comparing the expected values (smooth and progressive displacements) with outliers detected by the framework, it is possible to identify events that warrant more detailed technical attention for example, isolated records of anomalous displacements that may indicate reading failures or, in critical cases, unforeseen structural changes.

The implementation of Data Quality checks for the displacement fields *DESLOCRADIALABS* and *DESLOCTANGABS* could be decisive in assessing the seasonal movement of the structure.

The use of machine learning algorithms could enable not only more robust and adaptive outlier detection, but also the ability to learn historical patterns of failures or common behaviours by relating different data sources.

These techniques could also be useful in supervised learning to support, for example, the auto-adjustment of parameters in the Threshold Validator module, through the implementation of dynamic thresholds that can be recalculated at predefined intervals.

The framework can be expanded to include specific phases of data processing, such as the application of business rules or even complex transformations. Integration with more advanced visualization systems or even with Machine Learning tools is also possible, since the outputs are consistently generated in *.csv* format.

Additionally, integrating with reporting modules could support data analysis and the correction or calibration of sensor data, especially for users with limited technical background.

Bibliography

Adhikari, D., Jiang, W., & Zhan, J. (2021). Iterative Imputation Using Ratio-based Imputation for High Missing Gap. *2021 International Conference on Intelligent Technology and Embedded Systems, ICITES 2021*, 59–64. https://doi.org/10.1109/ICITES53477.2021.9637107

Batini, C., Rula, A., Scannapieco, M., & Viscusi, G. (2015). From data quality to big data quality. *Journal of Database Management*, *26*(1), 60–82. https://doi.org/10.4018/JDM.2015010103

Bhatti, M. T., Anwar, A. A., & Shah, M. A. A. (2019). Revisiting telemetry in Pakistan's Indus basin irrigation system. *Water (Switzerland)*, *11*(11). https://doi.org/10.3390/w11112315

Byabazaire, J., O'hare, G. M. P., & Delaney, D. T. (2022). End-to-End Data Quality Assessment Using Trust for Data Shared IoT Deployments. *IEEE Sensors Journal*, *22*(20), 19995–20009. https://doi.org/10.1109/JSEN.2022.3203853

DAMA International. (2017a). *DAMA-DMBOK: Data Management Body of Knowledge* (2nd ed.). Technics Publications.

DAMA International. (2017b). *DAMA-DMBOK: Data Management Body of Knowledge* (2nd ed.). Technics Publications .

DAMA International. (2017c). *DAMA-DMBOK: Data Management Body of Knowledge* (2nd ed.). Technics Publications .

Ehrlinger, L., & Wöß, W. (2022). A Survey of Data Quality Measurement and Monitoring Tools. In *Frontiers in Big Data* (Vol. 5). Frontiers Media S.A. https://doi.org/10.3389/fdata.2022.850611

Fang, C., Wang, X., Li, J., Wu, L., Wang, J., & Gu, H. (2025). A Novel Intelligent Learning Method for Identifying Gross Errors in Dam Deformation Monitoring Series. *Water (Switzerland)*, *17*(2). https://doi.org/10.3390/w17020148

Feng, C., Xiaowei, L., & Jishuang, H. (2025). A Web-based Framework for Spatiotemporal Integrity Verification of Dam Safety Monitoring Data. *Journal of Web Engineering*, *24*(7), 1023–1044. https://doi.org/10.13052/jwe1540-9589.2471

Feng, X., Zhang, ; Xiaotan, Changsen Sun, ;, Motamedi, M., Ansari, F., & Asce, M. (2013). *Stationary Wavelet Transform Method for Distributed Detection of Damage by Fiber-Optic Sensors*. https://doi.org/10.1061/(ASCE)EM.1943

Frontera-Bergas, M., Gelabert, C., Alorda-Ladaria, B., Sanchez, R., & Isern, E. (2024). Flash Flood Station Design with Data Quality assessment for SmartCity Applications. *2024 IEEE Sensors Applications Symposium, SAS 2024 - Proceedings*. https://doi.org/10.1109/SAS60918.2024.10636380

Fu, Y., Peng, C., Gomez, F., Narazaki, Y., & Spencer, B. F. (2019). Sensor fault management techniques for wireless smart sensor networks in structural health monitoring. *Structural Control and Health Monitoring*, *26*(7). https://doi.org/10.1002/stc.2362

Giordano, P. F., Quqa, S., & Limongelli, M. P. (2023). *Value of information analysis accounting for data quality*. 59. https://doi.org/10.1117/12.2657933

Gleeson, K., Husband, S., Gaffney, J., & Boxall, J. (2023). A data quality assessment framework for drinking water distribution system water quality time series datasets. *Aqua Water Infrastructure, Ecosystems and Society*, *72*(3), 329–347. https://doi.org/10.2166/aqua.2023.228

Guggilam, V. V., & Sundaram, G. (2024). Spatio-Temporal Bi-LSTM Based Variational Auto-Encoder for Multivariate IoT Data Imputation. *International Journal of Intelligent Engineering and Systems*, *17*(3), 352–367. https://doi.org/10.22266/ijies2024.0630.28

Karkouch, A., Al Moatassime, H., Mousannif, H., & Noel, T. (2015). *2015 IEEE ACS 12th International Conference of Computer Systems and Applications (AICCSA)*. IEEE.

Karkouch, A., Mousannif, H., Al Moatassime, H., & Noel, T. (2016). Data quality in internet of things: A state-of-the-art survey. In *Journal of Network and Computer Applications* (Vol. 73, pp. 57–81). Academic Press. https://doi.org/10.1016/j.jnca.2016.08.002

Karkouch, A., Mousannif, H., Al Moatassime, H., & Noel, T. (2018). A model-driven framework for data quality management in the Internet of Things. *Journal of Ambient Intelligence and Humanized Computing*, *9*(4), 977–998. https://doi.org/10.1007/s12652-017-0498-0

Kim, S., Castillo, R. P. Del, Caballero, I., Lee, J., Lee, C., Lee, D., Lee, S., & Mate, A. (2019). Extending Data Quality Management for Smart Connected Product Operations. *IEEE Access*, *7*, 144663–144678. https://doi.org/10.1109/ACCESS.2019.2945124

King, T., & Schwarzenbach, J. (2020). *Managing Data Quality - A practical guide*. BCS, The Chartered Institute for IT.

Kuemper, D., Iggena, T., Toenjes, R., & Pulvermueller, E. (2018). Valid.IoT - A framework for sensor data quality analysis and interpolation. *Proceedings of the 9th ACM Multimedia Systems Conference, MMSys 2018*, 294–303. https://doi.org/10.1145/3204949.3204972

Liu, C., Nitschke, P., Williams, S. P., & Zowghi, D. (2020). Data quality and the Internet of Things. *Computing*, *102*(2), 573–599. https://doi.org/10.1007/s00607-019-00746-z

Mansouri, T., Reza, M., Moghadam, S., & Zareravasan, A. (2021). *IoT Data Quality Issues and Potential Solutions: A Literature Review*.

Miller, R., Whelan, H., Chrubasik, M., Whittaker, D., Duncan, P., & Gregório, J. (2024). A Framework for Current and New Data Quality Dimensions: An Overview. *Data*, *9*(12). https://doi.org/10.3390/data9120151

Nguyen, L. A., Kiet, P. T., Lee, S., Yeo, H., & Son, Y. (2023). Comprehensive Survey of Sensor Data Verification in Internet of Things. *IEEE Access*, *11*, 50560–50577. https://doi.org/10.1109/ACCESS.2023.3277545

Nishio, M. (2017). Quality evaluation of fiber-optic strain data acquired in long-term bridge monitoring. *Sensors and Materials*, *29*(2), 141–152. https://doi.org/10.18494/SAM.2017.1462

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *International Journal of Surgery*, *88*. https://doi.org/10.1016/j.ijsu.2021.105906

Richter, B., Ulbrich, L., Herbers, M., & Marx, S. (2024). Advances in Data Pre-Processing Methods for Distributed Fiber Optic Strain Sensing. *Sensors*, *24*(23). https://doi.org/10.3390/s24237454

Sargiotis, D. (2024). Data Governance: A Guide. In *Data Governance: A Guide*. Springer Science+Business Media. https://doi.org/10.1007/978-3-031-67268-2

Tang, J., Zhang, G., Wang, Y., Wang, H., & Liu, F. (2015). A hybrid approach to integrate fuzzy C-means based imputation method with genetic algorithm for missing traffic volume data estimation. *Transportation Research Part C: Emerging Technologies*, *51*, 29–40. https://doi.org/10.1016/j.trc.2014.11.003

Tawakuli, A., & Engel, T. (2022). Towards Normalizing The Design Phase of Data Preprocessing Pipelines For IoT Data. *Proceedings - 2022 IEEE International Conference on Big Data, Big Data 2022*, 4589–4594. https://doi.org/10.1109/BigData55660.2022.10020312

Teh, H. Y., Kempa-Liehr, A. W., & Wang, K. I. K. (2020). Sensor data quality: a systematic review. In *Journal of Big Data* (Vol. 7, Number 1). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1186/s40537-020-0285-1

Venkatesan, R., Senthilkumar, P., Vedachalam, N., & Murugesh, P. (2017). Biofouling and its effects in sensor mounted moored observatory system in Northern Indian Ocean.

International Biodeterioration and Biodegradation, *116*, 198–204. https://doi.org/10.1016/j.ibiod.2016.10.034

Weiß, I., & Vogel-Heuser, B. (2021). A metric and visualization of completeness in multidimensional data sets of sensor and actuator data applied to a condition monitoring use case. *Applied Sciences (Switzerland)*, *11*(11). https://doi.org/10.3390/app11115022

Appendix A – Studies found in Literature Review

This appendix presents studies found for data quality management within sensor networks, and Big Data environments. This literature encompasses 19 studies from 2014-2024.

Author	Title	DQ Frameworks	DQ Dimensions	DQ Error Detection Methods	DQ Fixing Error methods	Application
(Miller et al., 2024)	*A Framework for Current and New Data Quality Dimensions: An Overview*	Hierarchical Domain-Agnostic DQ Framework	Taxonomy of all established & new dimensions	N/A	N/A	Foundational Theory
(Frontera-Bergas et al., 2024)	*Flash Flood Station Design with Data Quality Assessment for SmartCity Applications*	Embedded IoT DQ Assessment Architecture	Completeness, Accuracy	Real-time firmware anomaly detection	N/A	Water (Flash Floods)
(Richter et al., 2024)	*Advances in Data Pre-Processing Methods for Distributed Fiber Optic Strain Sensing*	Data Pre-Processing Pipeline	Accuracy, Reliability	High-resolution anomaly identification	Algorithmic signal filtering and smoothing	SHM (Strain Sensing)
(Guggilam & Sundaram, 2024)	*Spatio-Temporal Bi-LSTM Based Variational Auto-Encoder for Multivariate IoT Data Imputation*	N/A	Completeness	N/A	ST-Bi-LSTM-VAE deep learning imputation	IoT Sensor Networks
(Gleeson et al., 2023)	*A Data Quality Assessment Framework for Drinking Water Distribution System Water Quality Time Series Datasets*	Water Distribution Time-Series DQ Framework	Accuracy, Completeness, Timeliness	Technical outlier detection algorithms	N/A	Water (Drinking Networks)
(Giordano et al., 2023)	*Value of Information Analysis Accounting for Data Quality*	Value of Information Framework	Accuracy, Reliability	N/A	N/A	SHM
(Tawakuli & Engel, 2022)	*Towards Normalizing The Design Phase of Data Preprocessing Pipelines For IoT Data*	Normalized Data Pre-Processing Pipeline	Standardization	N/A	Systematic data transformation/cleaning	IoT Continuous Data
(Byabazaire et al., 2022)	*IoT Big Data Model, Trust, and Data Quality*	Big Data Trust Model	Trust, Veracity	N/A	N/A	IoT Big Data

(Weiß & Vogel-Heuser, 2021)	*A Metric and Visualization of Completeness in Multi-Dimensional Data Sets of Sensor and Actuator Data Applied to a Condition Monitoring Use Case*	Completeness Visualization Tool	Completeness	Missing data gap identification	N/A	Industrial Condition Monitoring
(Adhikari et al., 2021)	*Iterative Imputation Using Ratio-based Imputation for High Missing Gap*	N/A	Completeness	Continuous missing gap measurement	Iterative ratio-based imputation	IoT Time-Series
(Fu et al., 2019)	*Sensor Fault Management Techniques for Wireless Smart Sensor Networks in Structural Health Monitoring*	Decentralized Fault Management Strategy	Reliability, Accuracy	Distributed algorithmic fault detection	Fault isolation and data management	SHM (Smart Wireless Sensors)
(Kim et al., 2019)	*Extending Data Quality Management for Smart Connected Product Operations*	Extended DQM Framework for Lifecycle Operations	Accuracy, Timeliness	N/A	N/A	IoT (Smart Connected Products)
(Bhatti et al., 2019)	*Sensor Data Analysis for Water Flow/Canal Operations*	Operational Telemetry & Forecasting Framework	Timeliness, Latency	Systematic outlier detection	Time-series forecasting	Water (Canal Operations)
(Nishio, 2017)	*Quality Evaluation of Fiber-Optic Strain Data Acquired in Long-Term Bridge Monitoring*	Long-term Quality Evaluation Protocol	Reliability, Stability, Accuracy	Signal deterioration and spectrum evaluation	N/A	SHM (Bridges)
(Venkatesan et al., 2017)	*Biofouling and its Effects in Sensor Mounted Moored Observatory System in Northern Indian Ocean*	N/A	Reliability	Biofouling physical impact assessment	N/A	Water (Oceanic Observatories)
(Batini et al., 2015)	*Evolution of Data Quality Issues… to Big Data*	Conceptual Map of Big Data Evolution	Accuracy, Completeness	N/A	N/A	Foundational Theory / Big Data
(Karkouch et al., 2015)	*Data Quality Enhancement in Internet of Things Environment*	DQ Enhancement Pipeline Overview	Accuracy, Trustworthiness	Statistical anomaly and outlier discovery	General data cleaning techniques	IoT

(Tang et al., 2015)	*A Hybrid Approach to Integrate Fuzzy C-Means Based Imputation Method with Genetic Algorithm for Missing Traffic Volume Data Estimation*	N/A	Completeness	N/A	Fuzzy C-Means clustering & Genetic Algorithms	Time-Series Sensors
(X. Feng et al., 2013)	*Wavelet Transform Applications for Structural Health Monitoring*	Signal Processing Methodology	Accuracy	Wavelet-based anomaly / damage detection	Signal noise extraction	SHM (Vibration and Strain)

Appendix B – List of files shared by LNEC

The full list of files shared by LNEC can be found in the appendix section on the public shared repository of this master thesis under *hesis_appendixes/appendix_b/appendix_b.md* file[3]

[3] Direct link: Appendix B

Appendix C – Configuration Files

The configuration files used during the implementation of the DQ Framework presented in this master thesis can be found in the appendix section on the public shared repository of this master thesis under *thesis_appendixes/appendix_c/* directory[4].

[4] Direct link: Appendix C

Appendix D – Application Layer

The different tables presented in this appendix describe the different Profiling, DQ Checks and Threshold Validator modules implementations. Here are presented the DQ Checks and threshold validations implemented, csv output headers, but also file naming conventions for the outputs from the Reporting and Visualization Layer. Full data can be found in the appendix section on the public shared repository of this master thesis under *thesis_appendixes/appendix_d/* directory[5].

[5] Direct link: Appendix D

Appendix E – Results

This section shows all the outputs obtained after the practical application of the DQ Framework using sensor 00000 that are not represented in Results chapter.

The results are the output of the command *python .\src\dq_framework\dq_framework_exec.py --instrument plumbline --mode all --id_list 00000 --year_list 2024* , executed in the root project directory. The command means that the data used is for instrument *plumbline* for *all* (*automatic* and *manual*) readings files, for the sensor *00000* only for *2024* year.

The outputs follow the DQ Framework execution steps, meaning Profiling module outputs first, then DQ Checks Outputs for both Completeness and Timeliness DQ Dimension, and respective visualizations from the Reporting layer. After, Threshold Validator outputs and respective Reporting layer visuals. There are also the outputs for the Radial and Tangential displacements for both reading types and at the end, the Management and Politics Layer Document used as guide to implement the DQ Framework.

Full outputs can be found in the appendix section on the public shared repository of this master thesis under *thesis_appendixes/appendix_e/* directory[6].

[6] Direct link: Appendix E

Appendix F – Management and Politics Layer Document Template (author proposal)

The Management and Politics Layer Document Template author's proposal of can be found in the appendix section on the public shared repository of this master thesis under *thesis_appendixes/appendix_f/* directory[7].

[7] Direct link: Appendix F

Appendix G – Management and Politics Layer Document Proposal

The Management and Politics Layer Document Template author's proposal of can be found in the appendix section on the public shared repository of this master thesis under *thesis_appendixes/appendix_g/* directory [8].

[8] Direct link: Appendix G