A Strategy For Data Quality Assessment In IoT-based Air Quality Monitoring Systems

Julio Hernan Buelvas Perez

Master's thesis to opt for the title of Master in Telecommunications Engineering

Advisors

Natalia Gaviria Gomez, Ph.D.

Danny Munera Ramirez, Ph.D.

Universidad de Antioquia

Faculty of Engineering

Master in Telecommunications Engineering

Medellin

2022

Citation	Buelvas Perez, 2022 [1]
Reference IEEE Style (2020)	[1] Buelvas Perez, J. H. "A Strategy For Data Quality Assessment In IoT-based Air Quality Monitoring Systems", Research Work, Master of Telecommunications Engineering, Universidad de Antioquia, Medellin, 2022.

Master of Telecommunications Engineering, XV Cohort.
Research Group of Applied Telecommunications - GITA.
Environmental and Engineering Research Center.

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia
Sistema de Bibliotecas

Engineering Documentation Center- CENDOI

Institutional Repository: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Cespedes.
Dean/Director: Jesus Francisco Vargas Bonilla.
Head of Department: Augusto Enrique Salazar Jimenez.

Universidad de Antioquia

Abstract

With the upcoming growth of IoT, which is translated into millions of interconnected devices reporting a high volume of data coming from heterogeneous sources (sensors), it is necessary to assess the confidence of the data in order to provide the system with trustable information that can be used to get real insights from the physical world and thus take proper decisions or actions over it. Having in mind that ensuring data quality is key to ease user engagement, acceptance of IoT services and large scale deployments [1], a new critical issue arises which is related to the quality of the data in IoT applications. In order to get a correct insight or interpretation of the physical world, IoT users and upper layers need to be provided with reliable data and also need to be able to judge whether the data is reliable or not. Moreover, different users and applications will have different requirements and conceptions of data quality, making of this a subjective matter that should be included in the analysis. In this research, we investigate the data quality term and how it has been treated over several studies and applications, aiming at the definition of a set of attributes and metrics to assess quality of data in the Internet of Things. We also investigate on how to integrate objective measurements and subjective perceptions of data quality, to provide a single index that informs about the data quality status of the system. Our approach is implemented in a software tool, which is evaluated on a synthetic dataset to test its awareness to induced data quality changes. The tool is also tested with a real dataset retrieved from the SIATA's citizen science system, an air quality monitoring application that can be encompassed within the IoT paradigm, and that is composed by more that 230 nodes deployed all over the Aburrá Valley in Antioquia, Colombia. The results show that feasibility assessing data quality and the importance of data quality awareness for an IoT application, as a way for it to take proper actions on the real world.

Keywords

Internet of Things (IoT), Data Quality (DQ), Data Quality Assessment, Data Quality Dimensions.

Education is not the filling of a pail but the lighting of a fire.
William Butler Yeats

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

IoT (Internet of Things), shortly defined as "a network of items—embedded with sensors—which are connected to the Internet" [4], is expected to grow at such a point that by 2025 it is estimated there will be more than 21.5 billion of connected devices, compared to the current amount of about 8.3 billion devices [5]. This translates into about a 3-fold of heterogeneous connected things gathering and transmitting a large volume of data from the physical world to the digital world. In part, this increase of IoT connected devices is boosted by upcoming network and technological developments like 5G, and cheaper processing and sensing capabilities embedded in things.

This IoT growth will in turn trigger the rollout of applications like smart cities, where data from different domains will be collected and collated to decide about how to improve the efficiency of city resources in order to ease the life of citizens. The data collected by things within the smart city are used for monitoring and decision making. Such decision making may involve acting and reacting over the physical world or taking actions upon some phenomena. For example, to detect dangerous exposition to airborne pollutants that damage citizen's health, IoT can be used to acquire data from different sources, such as robust stations, citizen science systems or particular air quality monitoring projects. The data is used to get information about current levels of gases and pollutants in metropolitan areas. This information needs to be precise and trustworthy because it will be helpful when acting upon hazardous levels of pollutants.

Poor data quality leads to bad decisions and actions on the physical world, and it is noticeable that such decisions impact on people's lives. For example, authorities will create wrong policies and recommendations for citizens. In a more global view of a smart city, the authors in [6] note that imperfect information can have an adverse effect over the performance of urban services and decision making, it is necessary to deal with quality of data to improve the efficiency of smart cities since they seek to optimize the use of limited resources. From [7], two examples can be cited about the importance of data quality in different data mining contexts: the first example is related to quality of biomedical data, genomic data and their experimental results. When scientists collaborate with each other; they need to know the reliability of data if they will base their research on that data, because following incorrect theories and experiments will costs time and money. The second example is related to businesses and technological intelligence, where data is gathered from heterogeneous sources, such as technical reports, human assets, transcripts, commercial documents, competitive studies, knowledge-sharing websites, newsgroups, etc.; here it is highlighted that malicious or compromised data needs to be detected in order to avoid them from

influencing the decision making process and ensure critical decisions related to businesses. In [8], a study about vehicular network clouds integrated with IoT (VCoT) is presented. The authors highlight the importance of data quality to make reliable decisions based on the VCoT data; decisions range from simple actions, such as opening the garage door upon a car's arrival, to more critical processes like route calculation for an ambulance to the nearest hospital, or controlling traffic lights.

Because of the IoT nature, its data is exposed to many endangering factors [9], [1], and there is a huge concern on the data reliability and trustworthiness, e.g., using low-cost sensors is increasing the unreliability of data in IoT applications. Therefore, the data quality (DQ hereafter) needs to be studied, measured and enhanced in order to gain information, discover knowledge and propose solutions to react on the real world (known as the data lifecycle in IoT) [1]. In addition, it is worth to be mentioned that ensuring DQ is key to ease user engagement, acceptance of IoT services and large-scale deployments [1].

Data can be treated as a product and this is the approach taken by authors in [10] and [11] when trying to define a conceptual framework for data quality. The data consumer is in charge of assessing the quality of a product and deciding whether it fits its needs or purposes. In a general understanding, the term quality has been defined both as "fitness for use" and as "conformance to requirements" [9], [10], [12]. In the context of IoT, we can apply this concept, such that the application or user will have to decide whether the gathered data complies with the requirements. In order to understand and find a way to evaluate DQ from a customer perspective, [10] also defines a group of categories for Data Quality (Intrinsic, Contextual, Representational, Accessibility) and a set of attributes (hereafter dimensions) that belong to these categories, e.g. accuracy, completeness, timeliness, among others. These definitions, however, obey to information systems and databases contexts. In [1] and [9], we find a set of DQ dimensions that are related to IoT. However, among the reviewed literature, there is no agreement in terms of the group of dimensions that describe DQ, and more explicitly in the IoT context. Dimensions are named differently and/or used under a different meaning. Authors in [9] make an extensive study of the DQ in IoT, and selects a group of 6 dimensions that absorb other dimensions that have the same meaning or describe the same attributes.

Through an extensive review of different studies, we have compared the dimensions, metrics (a measure that gives quantitative assessment of a certain dimension) and methods used to estimate DQ, and we have found that even though DQ has been recently studied, to the best of our knowledge, there is not a standard or comprehensive way or methodology to assess DQ in IoT over its defined dimensions. Studies are limited to some dimensions, and most of them are usually focused only on accuracy [9], however, there might be applications which concern is more about completeness, timeliness or concordance, etc. In this way, the following research questions came up to mind:

- Which key parameters and dimensions should be taken into account to comprehensively assess the quality of data in an IoT application?

- How to estimate the quality of data in an IoT application?

- What if we could compute a data quality index to inform applications (or its users) about the feasibility of using that data to make proper decisions?

In order to answer these questions, in this work, we propose to define and test a strategy to assess DQ in IoT over a set of very well defined dimensions. In this way, the applications or users can be aware of data reliability through a DQ index which can be used to choose on making a decision or not upon the received data. Even though we think that our approach can be extended to many IoT applications, to demonstrate our approach we narrowed the scope to an IoT-based Air Quality Monitoring application composed of low-cost sensors. To this end, we built a model for the evaluation of data quality in that application. To develop the model, we investigated the main DQ dimensions in IoT systems, their metrics, and the DQ dimensions relevant to air quality monitoring. The model involves using the Pairwise Comparison Matrix-PCM technique to extract the subjective preferences of a data consumer regarding the attributes of the data product in a given context. In our case, the data consumers are experts users in air quality domain, however, any person at any domain can have different preferences for DQ in the given application. The model is a linear-weighted average that computes the DQ index of the system, as a function of the individual dimensions' DQ assessment, based on the defined metrics; and the set weights that come from the PCM results.

The model proposed as part of the strategy, was implemented in Python. The tool is flexible and escalable, and implements parallel programming techniques to improve the time performance of calculations, allowing to process large datasets in a short time. The tool uses APIs to automatically export the DQ results to Google Sheets and to Google Data Studio, where a web report is presented to the user. The report is composed by different views that inform about the system's DQ status. An overall report is provided, where the user can find the total DQ index. Moreover, other views present detailed information by DQ dimensions. That information involve tables, maps, histograms and time series that reflect the DQ behavior in different perspectives.

Preliminary results, about the feasibility of DQ assessment, were presented in [13], however, the tool was further improved for this research project. The model and the tool were tested in three different experiments that consisted on 1) Evaluating the time performance of the tool, 2) Verifying the DQ awareness of the tool by inducing changes in a controlled synthetic dataset, and 3) Assessing the DQ of a real dataset.

The main contributions of this research can be summarized as follows:

- The study, conceptualization and landing of DQ terms to the Internet of Things context.
- The identification and definition of DQ indicators and objectives in the context of air quality monitoring.
- The identification and proposal of metrics for an objective estimation of DQ, based on data and the system's contextual information.
- The estimation of an expert user's subjective perception of DQ importance, based on the Pairwise Comparison Matrix technique.

- The integration of a subjective assessment of DQ-dimensions' importance and objective DQ metrics to get a single DQ index.
- The design and implementation of a platform for a multidimensional assessment of DQ, including easy-to-read Python code in JupyterLab, multiprocessing to reduce the processing time in large datasets, an API to export data to a spreadsheet, and a web-based report to inform users about the DQ of the whole system as well as the time DQ evolution over the chosen period.
- The application of the platform to estimate the DQ of an air quality monitoring network.

The remaining of this document is organized as follows: In chapter 2, we present the DQ background and the state-of-the-art, which ends in the identification of DQ dimensions and metrics. In chapter 3, we expose the strategy to perform the DQ evaluation, which involves selecting the application and its dimensions, the model that integrate both subjective and objective DQ matters, and the output proposed to present the DQ results. Then, in chapter 4, we present the tool's modularized architecture, and we provide details about the implementation of each module. The test and results are presented and discussed in chapter 5. Finally, we present our conclusions and future research direction in chapter 5.5. In addition, we include three appendices that provide details about the PCM questionnaire (see appendix A), the generation of the synthetic dataset (see appendix B) and the user manual of the tool (see appendix C).

Chapter 2

Background and State-of-the-Art

In section 2.1 of this chapter, we present the data quality background and its definition, where we can see how this term has been approached not only in IoT, but in other contexts, and the research topics related to DQ. In section 2.2, a state-of-the-art is presented, emphasising in studies that treat the assessment of DQ in different contexts and IoT applications, and also, how they have used the DQ dimensions in their studies. Then, in the 2.3 section, the term *Internet of Things* is defined, as well as other solutions related to this paradigm. Finally, in section 2.4, we summarize the definitions and metrics of DQ dimensions in IoT, based on the state-of-the-art revision.

2.1 Data Quality

Data Quality is not a new topic, it has been widely defined and used in other contexts, apart from IoT, such as information systems. In [10], the author identifies the need to understand what data quality means to data consumers and after running two surveys on different data customers, defined a set of attributes that can be used to assess DQ. Data is treated as a product, which means that the data customers are the ones who define what is important to them. The result of this study is the identification of a set of 20 dimensions grouped in 4 categories: 1) intrinsic DQ, that consists of accuracy, objectivity, believability, and reputation; 2) contextual DQ, that consists of value-added, relevancy, timeliness, completeness, and appropriate amount of data; 3) representational DQ, that consists of interpretability, ease of understanding, representational consistency, and concise representation; and 4) accessibility DQ, that consists of accessibility and access security. The author states that this framework was effectively used in industry and government. For data quality definitions and studies, this work is one of the most extensively referenced ones. The same author proposes a Total Data Quality Management-TDQM methodology in [11], which is in the data domain, as the counterpart of Total Quality Management- TQM, which is in the product domain. The proposal looks to address and solve DQ problems that are caused due to the lack of grounded theoretical methodologies for TDQM. In TDQM, the author presents a methodology for managing DQ of an Information Product (IP), where four main tasks to manage DQ are defined: 1) Define IP, 2) Measure IP, 3) Analyze IP, 4) Improve IP. This approach matches the findings of the survey presented by [9], where four research themes related to DQ in IoT were identified: 1) Definition (dimensions), 2) Measurement (algorithms and techniques), 3) Analysis (manifestations of DQ problems) and 4) Design and Development (addressing and enhancing DQ).

Similar efforts to analyze DQ in the IoT field are found in the state-of-the-art survey [1]. In this work, the author defines DQ, DQ dimensions, the endangering factors that threaten DQ in IoT, the common manifestations of DQ problems, and defines the dimensions that are "fit for use" when assessing DQ in IoT. At the end, the author focuses on anomaly detection as a major problem of DQ, and finally provides a taxonomy of data cleaning techniques that are commonly used in IoT. However, this study lacks of methods for DQ assessment, which is mentioned as part of the open challenges and future research directions.

[14] provides a comprehensive study of data quality in the context of databases and information systems. The authors highlight the fact that data quality dimensions are the basis for any measurement and improvement of data quality. The assessment methodology process has three main activities: 1) relevant dimensions and metrics are initially chosen, classified, and measured; 2) subjective judgments of experts are performed; and 3) objective measurements and subjective judgements are compared. Different methodologies are compared and a new methodology is proposed: Complete Data Quality Methodology-CDQM. This methodology seeks to measure and improve data quality activities within an organizational context. This study also mentions that data quality in a dimension can be measured on different metrics and these metrics should be chosen according to their relevance to a particular problem, i.e., they are domain specific. The definition of data quality as "fitness for purpose" also arises since a database can be low quality for a given application but high quality for another one (for example consider an air quality application, where data should be at least 75% complete in order for it to be accepted according to a certain standard, however there is another regulation that states that data can be at least 50% complete), however, this consideration is thought to be a problem because it is not possible to have an objective assessment of data quality. Nevertheless, for dimensions like accuracy, completeness and consistency, it is possible to have an objective measurement which is evaluated based on the application's requirements.

Several studies focus on the analysis of data lineage [15–18], which in short terms refers to the documentation of the data life cycle in order to understand the origin and changes that the data have until they are stored in a database. That documentation is necessary so that researchers and their studies can rely on data obtained from secondary sources [15]. As mentioned by [16], the idea is that data is an asset instead of a liability. For that purpose, the authors proposed a framework to trace data lineage across several database technologies, in order to assure that the captured and maintained data is accurate, traceable, reliable and current, over time. In addition, having information about the data life cycle can assist the data quality measurement, since it is possible to study the source of data problems, and how each step that the data goes through impacts on the data quality. From those concepts, DQ dimensions like trust, accuracy and interpretability, can be related to the data lineage, which at the end looks for trustworthy and good quality data, so that informed decisions can be made.

2.2 Assessing Data Quality in IoT

This research focuses on the assessment of DQ in the context of IoT. When measuring DQ, the key point is to define the metrics [11] associated to DQ dimensions, thus first we consider literature

approaches that have identified certain dimensions that are commonly used and accepted in IoT, e.g., in [1], [9], [19], [20] it is evidenced that accuracy, timeliness, completeness, data volume, concordance and utility are frequently mentioned, being accuracy, timeliness and completeness the most studied ones. On the other hand, we delve into the studies presented in [9] to learn about the methods used to measure DQ. Following, we show some efforts found in the literature related to the assessment and improvement of DQ in different contexts.

In [21], a DQ study is performed in the field of Electronic Health Record (EHR) data, where they identify the need for a DQ assessment to determine the suitability of reusing EHRs for clinical research. To assess DQ, they first study the dimensions and then the methods that are used to measure DQ; five dimensions of data quality were identified, which are completeness, correctness, concordance, plausibility, and currency, and seven categories of data quality assessment methods: comparison with gold standards, data element agreement, data source agreement, distribution comparison, validity checks, log review, and element presence. The study done in [9] also groups the methods used to measure DQ into seven categories, namely: measurement between techniques, sources or defined attributes (MTS), measurement with a reference (MR), devices or algorithms validation (DAV), measurement between time intervals (MTI), measurement of presence (MP), process observation (PO), log files review (LR). Most of these groups of methods match between both studies, evidencing that in two different studies, there is an "agreement" about how to measure DQ.

The authors in [22] propose an automatic outlier detection technique where a model is built based on spatial correlation of weather variables as well as positional variables using Multiple Regression at each instant of time. Once the model is obtained from the training sample, it is then tested on the testing sample. Statistical measures, such as the standard error and the R-Squared, are used to check how well the model fits to the data. If the value is not within a given threshold, then the degree of the polynomial is increased and a new model is trained and tested. Once the model is found, it is used as a reference to compare against the real data. The values under a user-defined threshold are considered correct, while those off the threshold are considered outliers. The technique is successfully tested by identifying outliers on weather data. The authors claim that this method can be used for sensor readings on different domains. A similar approach is presented in [23], proposing three types of Multiple Regression models for the data: Linear, Polynomial and Generalized Additive Model-GAM. The number of predictors (variables) required for each type of model is decided using the elbow "trick", i.e., the Mean Squared Error-MSE is plotted against the number of variables used to build the model and the optimal number of variables is chosen where the MSE levels off. The number of predictors is increased in each step by using FSS (Forward Stepwise Selection) for each model, and the MSE is calculated for each of them. The results showed that GAM has a better performance in both static and streaming data. The performance is based on R- Squared and MSE metrics. The coefficient of determination tells us about how well the model fits the data and it is useful when comparing models, while the MSE, calculated as an average over all data points, tells us about the accuracy of the prediction.

The study [24] claims that most of the research is performed for database metrics (just to clarify, we call them dimensions, while this study refers to them as "metrics"). Metrics, such as precision, accuracy, and resolution, are objective and observable. However, the probability of correctness is not feasible in practice because the real value of a variable is not obtainable in a

pervasive environment. The study focuses on the feasibility and the utility of three basic metrics that are commonly used in pervasive applications: currency, availability and validity; these metrics have parameters that are interpretable and that can be obtained by analyzing historical data. Currency and availability depend on an expiration timer which is tunable, while validity is a set of rules that the data need to satisfy in order to be valid, e.g., a rule could be "the temperature of a certain city is between 0° C and 35° C", if the measured data is within this range, then the rule is met and the data is valid. The DQ of the mentioned dimensions is measured in real-world data to demonstrate their feasibility. The results are interpreted and discussed, and are found to be as expected for the behavior of the measured variables.

A Random Forest Regression model to ensure DQ over a set of weather data is proposed in [19], where DQ challenges (dimensions) in the Internet of Things are identified. This study focuses on accuracy, which is calculated using the Mean Absolute Percentage Error (MAPE), but the authors also mention that timeliness and completeness are left for future works. The accuracy is first calculated without a data cleaning processes using a baseline model, then it is calculated again after the data cleaning process. The experiment results show that the accuracy improves about 38.8%. From this study, it can be noted how important it is to clean the data before doing any analysis on it.

The research on log data, as part of the application's context, to assess DQ is proposed in [25], where the authors present a qualitative study about the adoption of WSN to monitor pharmaceutical cold chains in order to prevent loss of high value shipments, based on interviews to actors in the supply chain. By analyzing a case study, they show WSN can effectively improve the process and reduce wastes in the cold chain. Even though DQ is not studied, it is mentioned that by studying the log files of errors and claims investigation, they can analyze why pharmaceuticals perish due to temperature deviations, and the specific moments when these problems are most likely to occur; this translates in a way to find out anomalies based on logs, showing the importance and the impact of these logs' DQ. In addition, the authors mention how the accuracy of the temperature readings is affected based on the location of the sensors. This tells us about the importance of the context, which can be obtained from process observation.

In [26], authors propose a technique that automatically discovers the sensor feeds that best match users input of data and data quality requirements to the DQ properties of individual sensors. The system outputs the data in real time and automatically updates workflows according to queries and according to the availability of new feeds that better match the user's requirements. The results are evaluated in terms of the CPU usage reduction, data streams reduction, and bandwidth optimization. Although this technique is based on data that fits quality requested by customers (conformance to requirements), there are no calculations of DQ. The technique assumes that this information comes as metadata from the sensor and the system.

Authors in [27] are concerned about the reliability of sensors in the collaborative Internet of Things, since sometimes there is no description of the provided data and its precision. Thus, the authors propose a method for selecting sensors based on the correlation between the sensed data and a reference value. In this case, the data is obtained from open temperature sensors found in Xively, while the reference temperature is obtained from a public weather forecast service. The correlation coefficient and the MSE are used to measure accuracy. The sensor selection is done based on the correlation and a set of validity rules, as mentioned in [24]. The method was tested

on different types of sensor data and shows effectiveness in the proper selection of reliable sensors.

In [28], a prototype of a wireless multimedia sensor node capable to transmit at 3 Mbps (in a point-to-point topology) was developed, where the least acceptable rate to transmit video was calculated as 1 Mbps. It is comparably higher to Zigbee based platforms (widely used in WSN), that transmit at 250 Kbps. Even though the study does not show measurements of data quality, some metrics can be inferred to ascertain the resulting quality of the video: BER (the higher the bit error rate in the channel, the lower the quality of the video), channel transmission rate (the higher the transmission rate, the higher the quality of the video), user criteria (can objects in the video be clearly identified?). These metrics can be attributed to dimensions like data volume and completeness.

In the context of Mobile Crowd Sensing-MCS, authors in [29] propose a collaborative reputation score metric, as an alternative to statistical and vote-based user's reputation scores, to quantify crowd-sensed data trustworthiness. Reputation is associated to the mobile user, while trustworthiness is associated to the collected data, this means that the data trustworthiness is a direct consequence of the user reputation, hence the authors focus on the study of user reputation since it is directly proportional to the trustworthiness of that user's data. User reputation is composed of two metrics: Hard reputation, which is related to the accuracy of sensors, and is predictable since it comes from hardware based errors and can be detected; and Soft reputation, which is related to the probability of inaccurate readings that arise from malicious intelligence, such as a malicious user or a virus causing wrong reports, it is unpredictable. The authors claim that soft reputation involves the trustworthiness of the data, while hard reputation involves the correctness of the data. By simulation, the study compares the performance of three models, namely: 1) Trustworthy Sensing for Crowd Management-TSCM, 2) Collaborative M1, and 3) Collaborative M2, all based on 3 metrics: user utility, platform utility and false payments. The results show that the proposed method of collaborative reputation scores is an effective way to evaluate data trustworthiness in MCS. This approach could be extended to the assessment of related DQ dimensions in IoT, e.g. accuracy, utility, correctness and trustworthiness.

Also, targeting the MCS paradigm, [30] presents a cross validation (CV) approach to improve data quality in this kind of networks, through a validating crowd who ratify the sensor data provided by the contributing crowd (interim belief). The validation result reshapes data into a more credible posterior belief of the ground truth (measured sensor data). The approach is basically divided into 4 steps: 1) Profiling and Sampling: a profile consists of representative values and the probability distribution of those values. Then, the method implements a weighted random oversampling (WRoS) which consists of 2 configurations to represent the data to validators: reverse sampling, inverse sampling, and 2 more configurations for comparison: uniform sampling and proportional sampling. A single representative value is sampled to be shown to the validator with a probability proportional to its weight, which depends on the configuration. 2) Quest for validation: a validating crowd is recruited to assess the sensor data. 3) Reshaping or consolidation: the results of the validation and the original data are consolidated, and the reshaped data is represented by a posterior belief of data; 4) Compensation: a reputation is given to validators, and a payment is given to contributors. The results show that compared to other three configuration, reverse sampling performs good enough for both reinforcement of obscure truth and discovery of hidden truth, with up to 475% improvement of data quality, i.e., a given measured is ratified by the

validating crowd and then in the reshaped data (distribution), its probability increases. The change of belief (interim to posterior) is characterized using the Kullback–Leibler divergence. The measure can tell us how similar or different are the 2 probability distributions and could help in the assessment of accuracy.

In [31], a cross-layer approach is presented to assess DQ in the context of Wireless Sensor Networks-WSN. The WSN is composed of sensing nodes, aggregating nodes, sink nodes (base stations) and verifying nodes. The cross-layer validation consists of two parts: 1) sensor node location, that is calculated using verifier (or anchor) nodes' positions; and 2) data aggregation, that consists in adding data from multiple sensors into the aggregating nodes. Based on the trustworthiness of the sensing node's location, the sensor's trustworthiness data quality is qualitatively assessed as Malicious, Unknown or Robust, thus the application can decide whether to discard or to keep the data based on this metric. According to the study, when aggregating the data, the node can check if there are values out of range in comparison to close sensing nodes' measurements, i.e., it is about checking the lack of consistency. This can lead to the discovery of anomalous behaviors. In addition, data is encrypted from the sensing nodes to the sink node, passing by the aggregation nodes where data can be aggregated thanks to homomorphic encryption schemes that allow arithmetic operations; such approach guarantees data confidentiality and integrity. A simple use case is proposed to illustrate the method, however, there is not a comprehensive evaluation of it.

In [32], the author proposes a cloud-based solution that implements an Adaptive Sensing algorithm with Belief Propagation protocol (ASBP), to infer missing data and optimize the sensor selection by turning on only a small number of sensors. The author takes into account the problem of missing data due to nodes and links failures and uses the link quality as well as spatio-temporal correlation to minimize the power consumption. The BP (Belief Propagation), which is a technique used for solving inference problems, provides an iterative algorithm to infer sensor nodes measurements by incorporating the beliefs of other sensor nodes. In addition, CP (Constraint Programming) and greedy algorithms are used by the author to approach the optimization problem of sensor selection versus the required data quality. The approach is tested with real environmental sensor data from the Intel Berkeley Research Lab [33] and the results show that while keeping a good level of data quality (5 percent error in inferred data), the ASBP can help to save up to 80 percent more energy than when having all sensors activated. This study provides measurements for data utility and accuracy.

In [6], the authors study the concept of imperfect data (e.g. imprecision, uncertainty, ignorance, ambiguity, and/or incompleteness) and its impact on performance and decision making within a smart-city context. This issue is tackled by means of the theory of belief functions. The theory of belief function is claimed to be a powerful tool for modelling all kinds of imperfection and it is flexible to take into account the imperfection of data in pattern recognition and information fusion. The experiment was done on crowd-sourced healthcare data and the result was an Evidential Database (EDB), which includes perfect and imperfect data, that maps data to their evidential values. Evidential values are calculated based on the certainty degree of the source and the data consistency and can take characteristics like perfect data (certain and precise), Bayesian (precise but uncertain), consonant (imprecise but certain).

From studies like [34], it can be seen that most of them start by highlighting the problem of

sparse-fixed monitoring stations being insufficient to provide data about the spatial distribution of pollutants. They also mention that even though dispersion models can address this problem, their accuracy is limited. Low-cost sensors can be used to implement supplementary networks to increase the temporal and spatial resolutions of such high end monitoring systems. However, there is a high concern about the quality of data gathered from low-cost sensors. Informing users about the performance of a sensor is crucial for keeping desired levels of DQ. Authors in [34], assessed the performance of commercially available air quality sensors to provide scientific guidance to end users in choosing proper sensors by matching their requirements with the sensor's performance. This performance is closely associated with the quality of data, since depending on the sensor's capability to measure a variable, a good or a bad data will be acquired by the system. Regarding the Particulate Matter-PM sensors, the set of performance characteristics are the following: comparisons with reference measurements (reported with coefficient of determination R-Squared); repeatability and reproducibility characteristics (reported with the coefficient of variance CV and R-Squared), limit of detection (LOD, which values are lower that the EU concentration reference values); and dependence on particle composition, size (demonstrated to be highly dependant), humidity, and temperature (even though more research should be done, some studies have demonstrated dependence). Authors also mention the sensor stability, as the capability of the sensor to maintain its performance characteristics for a long time. Based on these characteristics (and here is where the end user plays a role) the study provides recommendations to end users and makes them aware of the DQ expected from a sensor under laboratory and field conditions. As pointed out, they also recommended performing sensor calibration under real conditions in order to improve the quality of the measurements. It is also important to include the aging/dust accumulation (sensing response changing over time) for better sensor calibration. In this case, end users can make a choice about what sensors to use according to their DQ.

2.3 Internet of Things Paradigm and Applications

Since its introduction in 1997 [4], the Internet of Things (IoT) paradigm has embraced different technologies. In the literature, we find some terms that are closely related to IoT, such as Wireless Sensor Networks (WSN), Pervasive/Ubiquitous Computing, Mobile Crowd Sensing (MCS), Cyber-Physical Systems (CPS), among others. IoT exploits these underlying technologies and paradigms with internet protocols and applications to make smart objects from traditional objects [35], and achieve automatic decision making. In [4], we can find several definitions, but the most basic definition for IoT was proposed by the IEEE in 2014: "A network of items—each embedded with sensors—which are connected to the Internet". In [35], a full review of IoT enabling technologies, protocols, and applications, is presented. In this review we can find a similar definition as "physical objects connected to the internet", linked to the fact that "IoT enables physical objects to see, hear, think and perform jobs by having them talk together to share information and to coordinate decisions". The basic IoT architecture is composed of three layers: Application layer, Network layer and Perception/Sensing layer, as shown in Figure 2.1. The application layer is related to storage and processing, and some terms like Big Data, cloud computing and Fog/Edge computing, have been used to address automatic decision making and volume data

processing. For the network layer, technologies like 4G, 5G, LoRA, WiFi, Zigbee, Bluetooth, etc., are involved. Finally, the perception layer is composed of smart things and devices with sensing and/or actuating capabilities.

2.3.1 WSN

Wireless Sensor Network is a spatially distributed network of autonomous sensors that monitor physical or environmental conditions, such as temperature, sound, pressure, etc. [4]. The WSN is composed of nodes connected to one or more sensors, and its scope is merely the collection of data. In contrast to WSN, IoT adds smartness to the objects so that they can do actuations to achieve some goal without human intervention.

2.3.2 CPS

In [4], the Cyber-Physical System is defined as a system of collaborating computational elements controlling physical elements. CPS goes to the level of networking between objects and sharing information about specific conditions in order to accomplish a certain goal with better efficiency. A smart grid is considered a good example for a CPS, where the behavior of consumers and suppliers is monitored to improve the efficiency, reliability, and sustainability of power production and distribution. From the networking point of view, IoT has a broader and global view of interconnected objects, usually to the internet, while CPS is related to the coordination of networked objects to achieve a specific goal within an intranet. Figure 2.2 depicts the integration of IoT and CPS, here the author sees IoT as the way to integrate different CPSs, however, as shown in Figure 2.1, IoT also encompasses the application and perception layers.

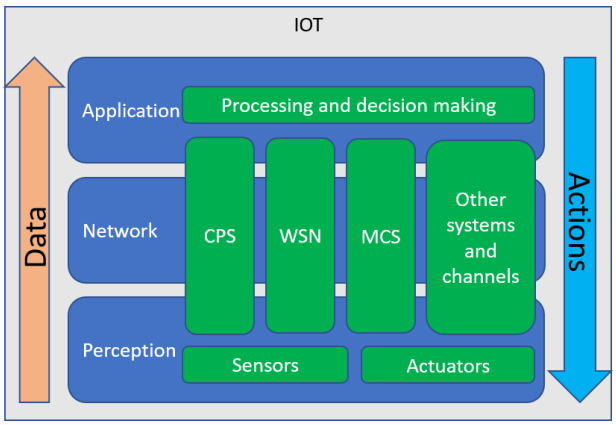

Figure 2.1: *IoT architecture and relationship with other systems.*

2.3.3 MCS

Mobile Crowd Sensing is a new concept, in which a central authority or platform and its participants (mobile users) work collaboratively to gather data from sensors embedded in smartphones,

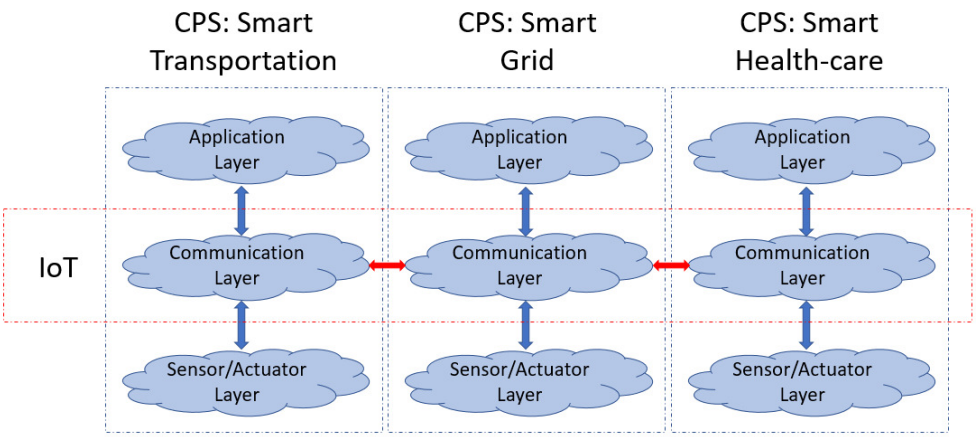

Figure 2.2: *Integration of IoT and CPS (taken from [2]).*

tablets, wearables, etc. [29]. In this system, there are data contributors who are given an incentive or compensation based on the quality of their data contributions. Some approaches [29], [30] propose data validators to ratify data provided by contributors. MCS enables a plethora of individual mobile phones and vehicular devices to share local knowledge collected by sensors (e.g., GPS, digital compass, microphone, light sensor, accelerometer, gyroscope, camera). The analysis of shared data provides useful insights for urban space monitoring that might have a great impact on society [36]. In relation to IoT, [30] claims that by leveraging personal sensing devices, MCS significantly accelerates the permeation of IoT as compared to the alternative of dedicated sensor deployment by governments and businesses. Thus, MCS becomes a key enabler of IoT by connecting things to cyberspace via the medium "humans-as-sensors".

2.3.4 IoT And Smart cities

Smart city is one of the most popular and targeted applications that can be leveraged by IoT, where the goal is to improve the quality of citizens' lives by using smart information infrastructure to ensure the sustainability and the competitiveness of different urban functions by integrating different dimensions of urban development and investments [6]. Smart cities have different key elements, such as smart grid, smart mobility/transportation, smart people, smart health, smart governance, smart economy, smart security, among others, that focus on different urban domains. E.g. smart security integrates hardware and software systems, sensors and mobile apps to read community surrounding conditions and combines other information sources to trigger alerts and recommendations in order to save citizens' integrity, which is endangered by external and internal threats, e.g., delinquency and natural disasters.

In smart transportation, citizens can determine how long it will take to get to a destination based on current traffic behavior. Also, real-time information about accidents on the road can lead to better traffic management. A proper traffic management within the city will also reduce the fuel consumption and enhance the air quality by reducing the pollution [37].

The smart grid is meant to replace the traditional power grid to provide reliable and efficient energy service to consumers. This grid design has some advantages, such as the improved utilization of distributed energy sources and resources, bidirectional networks, smart meters, allowing

reliability, efficiency, safety and interactivity [2].

By setting smart health applications, diseases can be diagnosed earlier, which leads to saving lives [38]. In smart health, citizens or risky patients can be continuously monitored to check their status, healthcare data can be accessed faster in case of emergencies, vulnerable populations can be identified, potential health risks can be predicted, etc.

Within the smart city, IoT helps to integrate such subsystems to improve the quality of services provided to citizens. This integration aims to achieve the best (efficient) use of public resources in cities [2].

2.4 Data Quality, Data Quality Dimensions And Metrics

The analysis of Data Quality (DQ) has been divided into dimensions, where dimension stands for an attribute that is important to the data consumer or the application. After studying the term DQ in the field of Internet of Things (IoT), we have identified several dimensions that are relevant within this context. In the following subsections, we define the most relevant dimensions that were identified. Also, we provide some metrics that can be used to estimate the DQ for each dimension. The metrics were constructed in such a way that the result is a number in the range $[0, 1]$, where values closer to 0 stand for poor DQ, while values closer to 1 stand for a good DQ.

2.4.1 Accuracy

Accuracy is probably the most important and studied dimension, In [9] [39], it is defined as "the degree to which data has attributes that correctly represent the true value of the intended attribute of a concept or event in a specific context of use". [9] also define it as "the extent to which an observation for the object truly reflects its real-world situation". The accuracy is related to other attributes, such as precision, validity, correctness and uncertainty.
[1] also provides a definition as "the maximal absolute systematic error α, such that the real values belong to the interval $[v-\alpha, v+\alpha]$", where $\alpha = \frac{|v_m - v|}{v}$, v_m is the observed or measured value, and v is the value accepted as true. It is evident that the accuracy is related to the similarity between the measured value and the real value. Equation 2.1 is the proposed metric for the accuracy dimension.

$$DQ_{accu} = \max(0, 1 - \alpha) \tag{2.1}$$

2.4.2 Precision

Regarding the precision, [40] defines it as "the degree to which further measurements of the same phenomenon in a close time instant provides the same or similar results" and it can be represented as the standard deviation of the measurement $\sigma = \sqrt{\frac{\sum_{i=1}^{n}(v_m - \bar{v}_m)^2}{N-1}}$, where $\bar{v}_m$ is the mean of the measurement over the N number of observations. To express it in the range $[0, 1]$, the coefficient of variation is used. Equation 2.2 is the proposed metric for the precision dimension.

$$DQ_{prec} = 1 - \frac{\sigma}{\bar{v}_m} \tag{2.2}$$

2.4.3 Timeliness

From [9] [39] the timeliness is described as "The degree to which data has attributes that are of the right age in a specific context of use". Another alternative definition is "The extent to which an observation for the object is updated at a desired time of interest" and it is related to terms like Currentness, Currency, Volatility, Latency, Freshness, Data rate, Delay, Frequency, Promptness. E.g. in [41] describes it as "The extent to which the age of data is appropriate for the task at hand". [1] provides a short and direct definition: "The difference between the current timestamp and the recording timestamp. May express both the age and the punctuality of a data item". It can be interpreted as whether the data is arriving on time to be used in the current tasks of the system. If the difference between current time and the arriving time of the data is off a defined range (if the observation is too old) the timeliness is lowered [42].

As proposed in [43], the timeliness can be calculated in terms of the $Currency = CurrentTime - Timestamp(v_m)$, and $volatility$, defined as the time during which data remain valid. Equation 2.3 is the proposed metric for the timeliness dimension.

$$DQ_{time} = \max\left(0, \frac{Currency}{Volatility}\right) \tag{2.3}$$

2.4.4 Completeness

Together with the accuracy and the timeliness related dimensions, the completeness is widely used in most of the studies. [9] [39] define it as "The degree to which subject data associated with an entity has values for all expected attributes and related entity instances in a specific context of use" and is related to attributes like the availability and missing data. An alternative definition is "The extent to which all expected data is provided by IoT services". In [1] the authors suggest "The ratio of non-interpolated items to all available (i.e. both non-interpolated and interpolated) data in the considered stream window". It is basically the ratio between the retrieved data and the expected data, as in equation 2.4.

$$DQ_{comp} = \frac{\#CollectedValues}{\#ExpectedValues} \tag{2.4}$$

2.4.5 Data Volume

Authors in [1] describe the data volume as "the number of raw data items (values) available for use to compute a result data item (in a stream query or sub-query)", while [9] give a similar definition "the number of data components that are transmitted from a source to a consumer for generating a data result". Data volume is different from previous dimensions since it is not defined as a "whether", "the degree to" or "the extent to". It instead is related to the amount of data required by the system to produce certain result. It is related and impacted by data loss (in compression of multimedia data), data transmission delay, data distortion, etc, and depends of the type of data, the network capacity and the operations that need to be done on data. For a fixed and low data transmission rate of a node, if the data volume is high, it will bring such problems. We can define it as the number of collected values retrieved at a time instant t, as in equation 2.5.

$$DQ_{dvol} = \#CollectedValues(t) \tag{2.5}$$

2.4.6 Data redundancy/Duplicates

Data redundancy or repeated data is accounted as for the amount of data items that have the same timestamp. This might be caused by abnormal network transmission that makes data to be transmitted or received multiple times [44]. The proposed metric for data duplicates is presented in equation 2.6.

$$DQ_{dupl} = 1 - \frac{\#RepeatedTimestamps}{\#CollectedValues} \tag{2.6}$$

2.4.7 Utility

According to [9], the utility is "the degree to which data can be accessed in a specific context of use", which is related to the data accessibility dimension. An alternative definition is presented by [9] and [45] as "the extent to which relevant data is accessed by data consumers from IoT datasets during a certain period of time" and it is related to terms like Usage, Frequency (of access) and Relevancy.

To calculate the utility dimension of DQ, it is necessary to keep track of user's or application's interactions with data in the form of queries or visualizations. It depends on whether data is provided by the system in "push mode", or in "pull mode" (i.e. queried by the user or application on demand), and proposed as in formula 2.7.

$$DQ_{util} = \begin{cases} 1, \text{ if data was accessed at least once in a period of time T} \\ \text{ or it is provided in push mode.} \\ 0, \text{ otherwise.} \end{cases} \tag{2.7}$$

2.4.8 Accessibility/Ease of access

Different from utility, [1] define ease of access as the availability and easiness of retrieving data, while the accessibility is regarded as a category of the DQ dimensions, and defined as "how accessible data are for data consumers".

In [43] it is calculated based on the time it takes to provide a result for a query. The metric is proposed in equation 2.8, where $RequestTime$ is the time when the query was launched, $DeliveryTime$ is the time when the answer was received, and $DeadLineTime$ is the maximum time at which the data is expected to arrive.

$$DQ_{acce} = \max\left(0, 1 - \frac{DeliveryTime - RequestTime}{DeadLineTime - RequesTime}\right) \tag{2.8}$$

2.4.9 Concordance/Consistency

In [9], the authors define Concordance as "the degree to which data has attributes that are free from contradiction and are coherent with other data in a specific context of use", it is related

to concepts like Consistency. An alternative definition cited by the same author and given by [42] is "the extent to which the data elements from a data source are in an agreement with the data elements from further individual data sources that report correlating effects". Both definitions take data from several sources to compare them in terms of correlation to evaluate their concordance.

Additionally, in [44], three different types of consistency are given: Consistency of acquisition frequency, Consistency of zero value and Instrumental consistency. The first one is related to the timeliness. The second one is related to the concordance of the zero value of one sensor with the value of other sensors in the same node. The third one is related to the similarity of an observation when measured with different instruments.

$$q_{con}(x_0) = \sum_{i=1}^{N} \lambda_i(x_0) \cdot c(x_0, x_i) \tag{2.9}$$

$$DQ_{conc} = \frac{q_{con}(x_0)}{N} \tag{2.10}$$

Based on the equation 2.9 proposed by [42], we can define 2.10. Where, with some modifications, $c(x_0, x_i) = |\rho_{ij}|$ is the absolute value of the Pearson correlation coefficient between variables x_0 and x_i, N is the number of variables, and $\lambda_i = \max\left(0, \frac{D-d_i}{D}\right)$ is a weight function that penalizes the correlation with variables according to their distance d_i. D is to be defined according to needs, e.g. it can be tuned as twice the distance to the nearest neighbor. It represents the maximum distance to take into account the correlation, where the weight is equal to zero.

An easier proposal is to calculate it as the absolute value of the Pearson's correlation coefficient between variables x_0 and x_i, as in equation 2.11:

$$DQ_{conc} = |\rho_{x_0 x_i}| \tag{2.11}$$

2.4.10 Validity/Plausibility

In [24], validity is a metric to evaluate the correctness of an observation, i.e. it is considered correct if it can be estimated, with a level of confidence, that the observation is within an acceptable range. It can be seen as a set of rules or constraints that data should comply with to be correct. While [42] defines plausibility as whether a received data source information makes sense regarding the probabilistic knowledge about what it is measuring. Both definitions are similar and variable ranges as well as historic information might be used to assess the validity of a measurement.

Suppose the following validity rules related to the correctness of the data:

- VR_1: Data is within allowed range.
- VR_2: Data consistency is greater than 90%.
- VR_3: Data accuracy is greater than 90%.
- VR_4: Data precision is greater than 60%.

Thus, as proposed by [24], the validity can be calculated as a series *and* operations over the m validity rules of a observation o, as in equation 2.12

$$DQ_{vali} = \bigwedge_{i=1}^{m} VR_i(o) \tag{2.12}$$

2.4.11 Interpretability

The interpretability tells whether data is clear in meaning and format [1], it can be improved by using annotations. According to [14], it concerns about the documentation and metadata that are available to correctly interpret the meaning and properties of data sources. In an IoT environment, where there is a high volume of heterogeneous data sources, this information is valuable to correctly read data, especially if sources are added and there is no prior knowledge of their format. Equation 2.13 is the proposed metric for the interpretability DQ estimation.

$$DQ_{inte} = \begin{cases} 1, \text{ dataset is annotated, there is metadata or there is documentation.} \\ 0, \text{ otherwise.} \end{cases} \tag{2.13}$$

2.4.12 Confidence

Authors in [46] and [1], define the data confidence as "the statistical error ε, such that $[v - \varepsilon, v + \varepsilon]$ contains the real value with a confidence probability of p". With $\varepsilon = z \cdot \frac{\sigma}{\sqrt{n}}, n \geq 30$, the proposed metric for the confidence data quality dimension is presented in equation 2.14, σ is the standard deviation within the sampled interval, n is the number of samples, and z is the score corresponding to the confidence level p. [46] explains that it represents the statistical measurement error due to random environmental interferences, such as vibrations or shocks.

$$DQ_{conf} = 1 - \frac{\varepsilon}{\bar{v}_m} \tag{2.14}$$

2.4.13 Trust/Reputation

In [41], trust is defined as "the probability by which data are suitable to be included in a specific process providing value", it is associated with source reputation and reliability. The authors also mention that source reputation is the sum of two main factors, the content reputation and the owner reputation, where the former depends on the number of times the source fails to provide a good answer, while the latter depends on the history of the organization that owns the data. E.g, a node can be given a reputation based on the quality of its provided data.

According to [47], data sources build a reputation that becomes trustworthy over time and that must be constantly re-assessed, thus trust can be formed, improved or lost.

Our proposal for the calculation of the trustworthiness is presented in equation 2.15, and it depends on two variables: 1) the source reputation is given by the user (or the IoT system, e.g. the source is authenticated or not) and takes two possible values $\{0, 1\}$, where 0 stands for bad reputation of the source and 1 stands for a good reputation of it. 2) the other variable is based on the correctness, i.e. the validity of data provided by the source during a certain period, for example, the latest n samples in a 24 hours window. The μ parameter can be used to tune which

side of the equation is more important, whether the reputation or the correctness of data. It depends on the availability of information to decide whether to include one or both variables, and decide which is more important.

$$DQ_{trus} = \mu \cdot Reputation + (1 - \mu) \cdot \frac{\sum_{i=1}^{n} DQ_{vali}(i)}{n} \tag{2.15}$$

2.4.14 Artificiality

Data artificiality is proposed by [42], and it determines whether data originates directly from a hardware sensor, or whether data is estimated after the application of techniques, such as interpolation, aggregation, fusion, etc. Equation 2.16 presents the metric for the artificiality DQ dimension.

$$DQ_{arti} = \begin{cases} 1, \text{ real sensor data.} \\ 0, \text{ artificial data.} \end{cases} \tag{2.16}$$

2.4.15 Access Security

Access security is defined in [1] as securing data in order to protect its privacy and confidentiality. In [40] [41] it is also found to be related to authentication and integrity. Both confidentiality and integrity are to be preserved for the received information, privacy should be protected for the transmitting source and the source authentication should be robust. In essence, sensitive data should remain confidential and private from its generation at the source to its storage in a database. It can be done by encrypting data. Regarding data authentication, it is related to data integrity and source authentication; The IoT system can (and should) verify the origin of the data and confirm its integrity (it is not corrupted or altered). If such mechanisms exist, these attributes can be evaluated as 1, or 0 otherwise. An example of such mechanisms is the use of cryptographic protocols, such as TLS.

Another approach, proposed by [41], for evaluating access security metrics, is based on identifying attacks and countermeasures for these attacks. Their system, named Networked Smart Objects-NOS, evaluates the robustness of the countermeasures to contain the attacks, and the system does it for each metric on run-time.

2.5 Conclusions

To conclude, in this chapter we identified key aspects of data quality in the context of IoT and key technologies that fall within the scope of IoT. As shown in the literature, such technologies also include DQ terms, which help us see the magnitude of this concept. In the literature, it was possible to find that DQ is a multidimensional concept which has been approached in different ways, and that involves all parts of the system, independently of its context or technology. Most of the studies focus on accuracy calculations, for which a reference is needed; sometimes that reference can be obtained from other sensors, user inputs or models. Some of the studies describe methods

DQ Dimension	Studies										
	[9]	[1]	[41]	[47]	[20]	[48]	[45]	[42]	[44]	[49]	[40]
Accuracy	✓	✓	✓	✓					✓		✓
Precision											✓
Timeliness	✓	✓	✓		✓	✓		✓	✓		✓
Completeness	✓	✓	✓					✓	✓	✓	✓
Data volume	✓	✓									
Data redundancy/Duplicates		✓							✓		
Utility	✓						✓				
Ease of access/accessibility		✓									
Concordance/consistency	✓							✓	✓		
Validity/Plausibility					✓			✓			
Interpretability		✓									
Confidence		✓									
Trust/Source Reputation			✓	✓							
Access security		✓	✓								✓
Artificiality								✓			

Table 2.1: *DQ dimensions present in IoT-related studies*

to evaluate DQ, some of them build models to estimate DQ, some others provide explicit equations to estimate it at some DQ dimensions, while some others just mention DQ dimensions. It is worth mentioning that the use of contextual information is a key point when evaluating DQ, it is because each application is deployed in a very specific context, where the DQ expectations are different, and the endangering factors are also different, thus having knowledge about the context will help to better study the DQ dependence of application factors. Even though some studies agreed on definitions, dimensions or DQ measurement techniques/metrics, we did not find a complete way to estimate DQ in IoT, and over all its define dimensions. Table 2.1 shows a summary of several IoT-related articles and the set of dimensions that were covered by them. Based on these findings, we identified a set of commonly used dimensions in the context of IoT, we provided their definitions, and proposed some metrics for their individual evaluation, which were directly extracted from the related studies, or were inferred from the definitions, and adapted to give a result between 0 and 1.

Chapter 3

Data Quality Evaluation Strategy

In this chapter we present the strategy for data quality assessment and how the result will be presented. First of all, in section 3.1, the target application of air quality monitoring is studied to understand and gather knowledge about what kind of data is provided and how large it is, what data quality endangering factors are present, what are the sensor specifications, among others. Then, in section 3.2, the set of data quality dimensions, also called data quality indicators, is identified based on the application's context, to continue with 3.3, where the proposal of metrics to evaluate the quality of data for each dimension are given based on a mapping between IoT dimensions and air quality indicators. In section 3.4, it is proposed a weighted average model to estimate the DQ index based on the selected dimensions and the user subjective preferences of data quality, that are estimated based on the Pairwise Comparison Matrix technique. And finally, the proposal to provide the result of the DQ assessment is presented in 3.5.

3.1 The Application

Several IoT datasets were considered for this work [33, 50–53], however, the target dataset was from the SIATA citizen science system, a science and technology project of the metropolitan area of the Aburrá Valley and the major's office of Medellín, whose objective is to identify and forecast the occurrence of natural and anthropic phenomena that alter the environmental conditions of the area, or that may generate risks to the population. To accomplish this goal a real-time monitoring system of hydrological and meteorological variables has been deployed, to timely delivery information to citizens, which added to educational processes and the development of community early warning systems, enabling the protection of life and the environment in the region [54].

We chose the SIATA's citizen science air quality monitoring system because it has the following characteristics that make it a good IoT application to be studied: 1) It is a city-scale deployment with more than 230 nodes and 22 robust stations covering more than 382 km^2 of the Aburrá Valley in Antioquia, Colombia, 2) Each node counts with two low-cost air quality sensors and one humidity and temperature sensor, 3) The robust stations provide reference data against which low-cost nodes data can be compared, 4) The nodes are located at citizens' premises, hence they rely on citizens' power grid and internet access, 5) The data is available online, but historical data can also be queried, 6) This is a local and public project, which makes it easier to get in touch with system's administrators to check on any doubts or clarifications, 7) System features like the

Sensor	Variable	Units	Sensor Range	Precision	Resolution	Period
Davis 6830	Relative Humidity	%	1% to 100%HR	2%	1% RH	1 minute
	Temperature	$°C$	$-40°C$ to $+70°C$	$0.3°C$	$0.1°C$	1 minute
HK-A5 Laser	PM2.5, PM10 from df sensor	$\mu g/m^3$	$0\mu g/m^3$ to $999\mu g/m^3$	50% at $0.3\mu m$, 98% at $\geq 0.5\mu m$	$0.3\mu m$	1 minute
Nova - SDS011	PM2.5, PM10 from nova sensor	$\mu g/m^3$	$0.0\mu g/m^3$ to $999.9\mu g/m^3$	Relative error: Maximum of $\pm 15\%$ and $10\mu g/m^3$	$0.3\mu m$	1 minute

Table 3.1: *Citizen Science Low-cost Sensors Specifications*

deployment scale, the sensor location and the fact that low-cost sensors are used, threaten the quality of the data, and make this systems challenging and at the same time interesting, 8) It is also important to count with the data provider's information as well as count with reference stations, which help to evaluate different attributes of Data Quality.

The gathered data is assigned with a data quality tag for each data point before being published online. The DQ tags are the result of taking into account aspects like national standards, equipment providers' recommendations, validity range according to the historical data and international guidelines (e.g., Quality Assurance Handbook for Air Pollution Measurement [55]). Robust SIATA stations, that will be used as reference, report a measurement every hour, while citizen science low-cost nodes, that are the focus of our work, report measurements every minute. The nodes are implemented with a Davis 6830 sensor for relative humidity and temperature measurements, and two sensors for particle matter (PM1, PM2.5 and PM10) measurements, the HK-A5 Laser (a.k.a. DF) and the Nova - SDS011 sensors, details are given in table 3.1. Note that to show our approach, in this work we will only analyze the PM2.5 measurements.

Regarding the number of sampling points, for zones like the Aburrá Valley in Antioquia-Colombia, with about 4 million inhabitants in 2020 [56], the minimum number of sampling points is 11 (the worst case) if the indications provided by the guideline [57] are followed. This zone has 22 fixed stations (providing fixed measurements, i.e., robust stations with high quality data) and about 230 nodes providing indicative measurements (according to the guideline, the indicative measurements are those with less strict data quality levels), thus complying with the established guidelines. Figure 3.1 displays the distribution of fixed and indicative stations and nodes in the mentioned region. In [13], it was shown that the maximum distance between a node and a robust station is around 7 km, while the minimum distance is in the term of a few meters, and the average distance is about 2 km. According to the study, there was not found a correlation between the distance and the accuracy, however it is expected that such distance will cause a bias between the measurements provided by both fixed and indicative measurements.

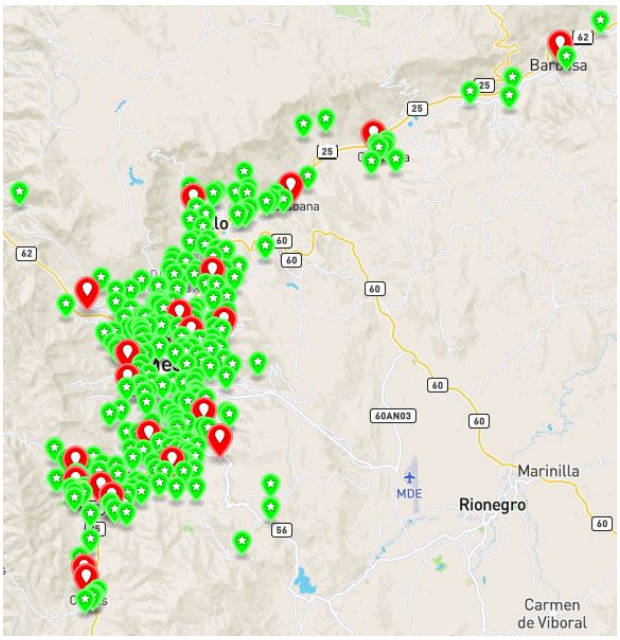

Figure 3.1: *Air quality monitoring stations in the Aburrá Valley (note that red spots are for robust stations and green spots are for low-cost nodes).*

3.2 Data Quality Indicators In Air Quality Monitoring Systems

Previously, in chapter 2.1, we identified and defined the IoT data quality dimensions, however, in the context of air quality monitoring systems, the important data attributes to evaluate quality of data are called Data Quality Indicators (DQI), and there are also Data Quality Objectives (DQO) that stand for the levels of accepted thresholds of such DQI. There are two main entities that defined the indicators and requirements, namely the *The European Parliament And The Council* in the EU with the *DIRECTIVE 2008/50/EC* [57], and the *Environmental Protection Agency (EPA)* in the US with the *Quality Assurance Handbook for Air Pollution Measurement Systems* [55]. From both guides, it was possible to identify and match some DQ indicators to the DQ dimensions previously discussed. This revision is required because users of air quality monitoring systems have different requirements of DQ than users of other IoT applications.

The DQ indicators are defined below. Some DQO will also be given if they are available in the guidelines. Note that metrics are not provided for the DQ indicators, however, in figure 3.2 a mapping between air quality indicators and data quality dimensions is given, which will help to define the metrics to be used.

3.2.1 Uncertainty

According to [58], uncertainty is "a parameter, associated with the result of a measurement, that characterizes the dispersion of the values that could reasonably be attributed to the measurand". Also, [58] states that uncertainty is a generic term used to describe the sum of all sources of error associated with an environmental data operation. Uncertainty has two components namely population uncertainty, which is related to the representativeness of the sample and measurement

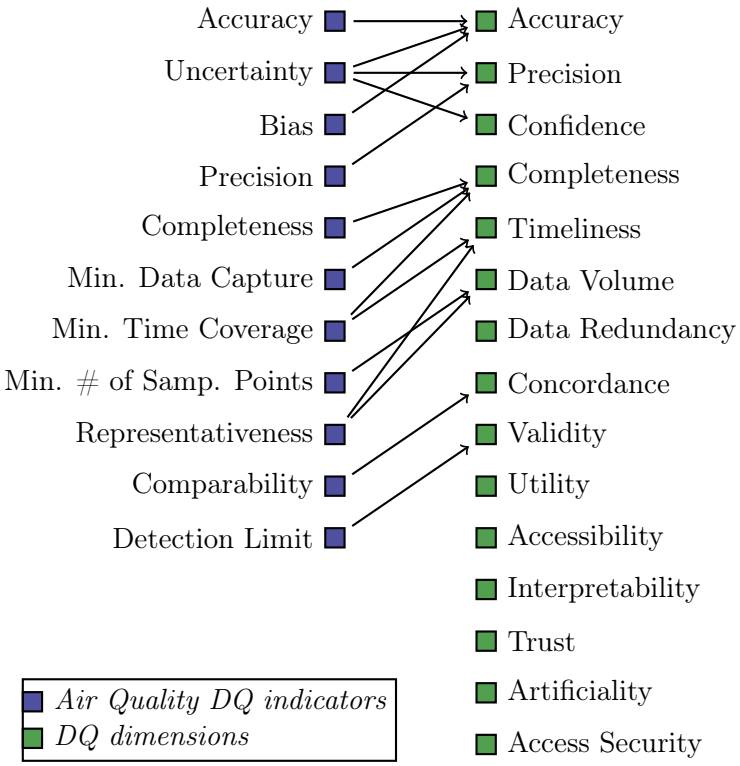

Figure 3.2: *Mapping Air Quality DQ indicators to DQ dimensions.*

uncertainty, which is related to the precision, bias and detection limit [55]. They will be defined below.

Regarding the DQO for particulate matter pollutants, according to [57], the maximum allowed uncertainty for fixed measurements (robust monitoring stations) is 25 %, while for indicative measurements (e.g. low-cost sensors measurements) is 50 %, meaning that the sum of error from the considered sources of error is more flexible for indicative measurements than for fixed measurements. This indicator is related to data accuracy and confidence.

3.2.2 Minimum Data Capture

This indicator is found in [57] and has a limit of 90%, meaning that the maximum number of missing values within one measurement period is 10% of the expected values. For example, in an one-hour period, where a sensor is reporting 60 measurements (a sample every minute), the minimum data capture is 54 samples. This indicator is related to completeness.

3.2.3 Minimum Time Coverage

This indicator is found in [57] and for indicative measurements of pollutants, such as particulate matter (PM10/PM2,5), it has a limit of 14% (one day's measurement a week at random, evenly distributed over the year, which would result on roughly 52 one-day measurements per year, or eight weeks evenly distributed over the year, which would result on roughly 56 one-day measurements per year). This indicator is related to timeliness and completeness.

3.2.4 Minimum Number of Sampling Points

This indicator is defined in [57] for fixed measurements. The amount of sampling points depends on a zone's population and area. This indicator is related to data volume.

3.2.5 Precision

Precision represents the random component of error and is a measure of agreement among repeated measurements of the same property, under identical or very similar conditions [55]. It is usually estimated as a derivation of the standard deviation. This indicator is part of the uncertainty components and matches the precision DQ dimension.

3.2.6 Bias

This indicator is a component of the uncertainty and represents the systematic distortion of a measurement process that causes error in one direction. It is determined by the estimation of positive and negative deviation from the true value [55]. This definition matches the accuracy DQ dimension.

3.2.7 Detection Limit

Also known as limit of detection, it is the minimum concentration of a pollutant that can be distinguished from zero (absence of the pollutant) by a single measurement at a stated level of probability [55]. This indicator can be sorted within the validity DQ dimension.

3.2.8 Accuracy

The accuracy as a data quality indicator is defined in [55] as "measure of the overall agreement of a measurement to a known value and includes a combination of random error (precision) and systematic error (bias) components of both sampling and analytical operations". The guide recommends using bias and precision when possible, otherwise use accuracy as the measurement uncertainty. This indicator matches the dimension of the same name.

3.2.9 Representativeness

In handbook [55], representativeness is a measurement of the population component of uncertainty and refers to "the degree to which data accurately and precisely represents the frequency distribution of a specific variable in the population". According to the guide, it does not matter how precise or unbiased the measurement values are if a site is unrepresentative of the population it is presumed to represent. Representativeness depends on factors like the amount of sampling points (network size), frequency of sampling and sampling schedule. Thus, this indicator can match timeliness and data volume DQ dimensions, it also matches the "minimum number of sampling points" and "minimum time coverage" that are discussed in the guide [57].

3.2.10 Comparability

In the EPA handbook [55], this indicator is defined as "a measure of the confidence with which one dataset or method can be compared to another, considering the units of measurement and applicability to standard statistical techniques". For example, if there are two datasets retrieved from monitoring stations and low-cost sensors, it is expected that both of them are comparable. This indicator can match the confidence DQ dimension if the measurements of one dataset fall within the confidence interval of the other. Also, it could match the concordance dimension.

3.2.11 Completeness

This indicator (from [55]) directly matches definition of the data completeness DQ dimension as the ratio of valid obtained data to the expected data. EPA requires 75 % data for it to be complete.

3.3 Metrics Selection

Using previous definitions and having in mind the application in the context of air quality monitoring, it is possible to map the AQ DQ indicators to general IoT DQ dimensions as in figure 3.2, this mapping helps to identify which air-quality data attributes are related to the IoT data quality ones. As evidenced, not all dimensions have a match on the indicators side. One probable explanation is that air quality monitoring applications only care about sensors' measurements, performance, and disposition. In contrast, the context of IoT is broader, and there is also concern about contextual and system aspects like the utility of data, its accessibility, interpretability, artificiality, accessibility, trust and access security.

The metrics or formulas for the assessment of air quality monitoring indicators are not explicitly given in the guidelines, and not being experts in air quality monitoring makes of it a non straightforward task to find them. However, starting from the definitions we could find similarities among both dimensions and indicators concepts, allowing us to propose the mapping. With this mapping, we can apply the metrics proposed in section 2.4 to evaluate the DQ of this application, helping to demonstrate of the approach exposed in this research. Metrics for DQ dimensions in IoT are much easier to be identified and are usually consistent among different works. The metrics to be used are the proposed ones and they are not mandatory, actually, a different set of metrics to assess the DQ levels can be used. For example, an expert in the air quality domain could redefine the metrics to fully comply with the guidelines definitions, or the metrics could be set to boolean $True/False$ results depending on whether or not an indicator complies with the DQO. The uncertainty indicator could be used to encompass the bias, accuracy and precision indicators by developing a metric that considers all the sources of error, however, this calculation would be more complex since identifying and characterizing all sources of error in an IoT application is a difficult task. It will be found in the following chapters that the uncertainty indicator was used, but for only one source of uncertainty, the *Between Sampler/instrument uncertainty* defined in [59] as the equation 3.1, where n is the number of samples, $y_{i,DF}$ and $y_{i,NOVA}$ are the measurements of DF and NOVA sensors respectively, and $\bar{y}$ is the average of both DF and NOVA measurements.

$$Uncertainty_{BS} = \sqrt{\frac{\sum_{i=1}^{n} (y_{i,DF} - y_{i,NOVA})^2}{2n\bar{y}^2}} \tag{3.1}$$

For consistency with other DQ dimensions, we can define equation 3.2 to get a value between 0 and 1, where 0 means a bad quality and 1 means a good quality.

$$DQ_{uncer} = \max(0, 1 - Uncertainty_{BS}) \tag{3.2}$$

3.4 Data Quality Expectations And Subjective Measurements

3.4.1 Preliminary Studies

The study [60] focuses on the evaluation of the Quality of Experience (QoE), which is based on a layered approach where parameters (influencing factors) belonging to each layer (i.e. object layer, network layer and application layer) are used to evaluate the QoE. The QoE for each layer depends on parameters that are intrinsic to the network (referred as QoS, involving delay and packet loss) and parameters that are intrinsic to the device objects (referred to as Quality of Data - QoD, such as accuracy and precision, bit rate, resolution, codec, among others). Furthermore, in the application layer, parameters like the presentation (user interface) and context are used as well for the QoE estimation. Through a defined set of test conditions, the researchers look to subjectively assess the overall QoE by calculating the Mean Opinion Score (MOS). For each test condition, some of the previously mentioned parameters are modified in order to expose the users to different scenarios. After the test is finished, the authors have a dataset of the IoT platform parameters (objective measurements) plus the MOS (subjective perceptions), which are used to build a model to estimate the QoE based on metrics from the different IoT layers. The model was built using the training part of the dataset, and evaluated using the validating part of the dataset, obtaining good Pearson correlation figures of above 93% in most of the cases.

It is worth noting that some terms used in QoE studies are similar to DQ terms in our research, and they could be used interchangeably. For example, according to [60], the QoE can be defined as "the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the user's personality and current state". Considering that most of the authors define DQ as dependent on user requirements, the assessment of both DQ and QoE are user-centred, which opens the door to evaluate the Quality of Data based on QoE techniques.

Similarly, in [61], the authors discuss the QoE in the IoT context, rather than in the multimedia context. In the latter, the communications are usually H2M (Human to Machine) or M2H (Machine to Human); in the former, however, the communications are usually M2M. From this point, the authors state that subjective measurements of QoE should be avoided in the IoT context, or at least modelled in terms of the usage (log containing installation/deinstallation, used time, most used features, etc) of the application by the user. The QoE has objective predictors such QoS (regarded as the technical part), the physical part which involves the objects' features, such as power, storage, computing capacities, among others; and finally the QoD (regarded as the contextual part), where the dimensions are mentioned. In this study, the QoD is also treated as

a multidimensional concept.

The work presented in [62] estimates the QoE of smart-wearables based on QoD and Quality of Information-QoI metrics. For instance, a quantitative QoE model is proposed as a weighted linear combination of parameters from QoD and QoI categories. Several methods for finding the weights are proposed: 1) A balanced weight distributions, where all parameters have equal weight, 2) A correlation based distribution, where the weights are calculated based on the correlation between parameter rating and the overall rating (both ratings were previously obtained using questionnaires to evaluate the MOS), 3) A hybrid distribution that consist on mixing methods 1 and 2, and 4. A priority based distribution, where users are asked to do a comparison of the parameters (i.e. Pairwise Comparison technique) that leads to the weights.

In [63], the proposed subjective DQ assessment is based on questionnaires answered by stakeholders, where the questions are done regarding the different dimensions of DQ. The same authors mention that the objective measurements can be task-dependent and task independent whether or not they need contextual knowledge, respectively. A contribution of this article is the presentation of functional forms for developing objective DQ metrics, namely Simple Ratio, Min and Max operation and weighted average. This study also mentions the use of subjective and objective DQ assessments to improve the DQ by identifying discrepancies between them and finding their root causes.

In [64], the authors also mention the objective measurements and subjective measurements of DQ. Dimensions are sorted on Primary Quality Criteria and Secondary Quality Criteria. Both primary and secondary quality criteria (dimensions) are related to the objective and subjective DQ assessment, respectively. The subjective assessment is done through the use of questionnaires. In this study, authors define and explain the metrics to assess DQ and explore the interdependence of some DQ dimensions, using some of them as the inputs to assess others.

We can conclude from the revision above that QoD can be modeled in terms of the underlying IoT physical objects features, network characteristics, contextual knowledge about the application and some subjective "inputs" from users. Furthermore, that questionnaires are used to get subjective estimations of DQ. The questionnaires are answered only one time when building the model. Once modeled, there will not be need for further subjective assessments, hence automating the DQ evaluation.

3.4.2 A Single DQ Index

In this work, we plan to use both a separated and an aggregated metrics to be available. The aim of using an aggregated index number is to summarize simple index numbers contained in a complex number (a value formed from a set of simple values) in just one value [65]. As highlighted by [63], a single-valued aggregate data quality measure would be subject to deficiencies associated with widely used indexes (like the Consumer Price Index), where many variables and weights would be subjective. A single value would also hide valuable information about specific issues related to DQ, that otherwise several indicators would make evident. However, if assumptions and limitations of the composite index are well defined, and the index is properly interpreted by the user, the bias produced by the subjective assessment can be controlled, and the index could

be useful to assess DQ. It will also make much easier to provide comparative assessments over time since, visually, a single indicator is easier to analyze and follow. The separated indexes can be used to investigate the DQ degradation of specific dimensions.

3.4.3 Subjective Measurement Of DQ

We propose a subjective measurement of DQ using a weighted linear combination of DQ dimension indexes to obtain the overall DQ index, similar to the strategy proposed in [62] for QoE, where authors used the Pairwise Comparison Matrix-PCM to get the user priorities, which are put in a weighted average equation as a model to estimate the overall QoE, but instead of QoD and QoI parameters, we use dimensions indexes. This is shown in equation 3.3. This equation allows to obtain an overall DQ index based on a set of DQ estimations for each dimension DQ_{dim}, and a set of weights w_i or v_j got from the PCM, this technique is discussed later.

$$\begin{aligned} DQI = \mu \cdot (w_1 DQ_{accu} + w_2 DQ_{prec} + w_3 DQ_{conf} + w_4 DQ_{comp} + w_5 DQ_{time} \\ + w_6 DQ_{volu} + w_7 DQ_{redu} + w_8 DQ_{conc}) \\ + (1-\mu) \cdot (v_1 DQ_{util} + v_2 DQ_{acce} + v_3 DQ_{inte} + v_4 DQ_{trus} + v_5 DQ_{arti} + v_6 DQ_{acce}) \end{aligned} \tag{3.3}$$

We use the parameter μ as mechanism to adjust the relative importance to data-related dimensions (accuracy, precision, confidence, completeness, timeliness, data volume, data redundancy and concordance) or system-related dimensions (utility, accessibility, interpretability, trust/reputation, artificiality and access security). The particular value of μ can be fine tuned by the user according to the particular context. In order to make the DQ index to lie within the range $[0, 1]$, it must be satisfied that $\mu \in [0, 1]$, and the weights on both parts must satisfy the condition that their sum is equal to one:

$$\sum_{i=1}^{8} w_i = 1 \tag{3.4}$$

$$\sum_{i=1}^{6} v_i = 1 \tag{3.5}$$

If the number of dimension to be considered is lower OR higher, both equations can be adjusted accordingly.

The Pairwise Comparison Matrix technique is part of the Analytic Hierarchy Process (APH), a multi-criteria decision making approach proposed by [3], that aims to identify the preferences that an individual has over a set of factors. We used it to evaluate the priorities that a data consumer has over a set of Data Quality dimensions in a given IoT Context: Air Quality Monitoring. Such preferences or priorities resulting from the PCM will be used as the weights w_i and v_i in equation 3.3.

To compare the factors, we use the fundamental scale of absolute numbers [66], [3]. It con-

sists of verbal judgments for comparing two factors using a range from equal to extreme (equal, moderately more, strongly more, very strongly more, extremely more), and corresponding to the verbal judgments are the numerical judgments (1, 3, 5, 7, 9) [3]. Table 3.2 presents the verbal judgements in more detail.

Intensity of importance of an absolute scale	**Definition**	**Explanation**
1	Equal importance	Two activities contribute equally to the objective
3	Moderate importance of one over another	Experience and judgment strongly favor one activity over another
5	Essential or strong importance	Experience and judgement strongly favor one activity over another
7	Very strong importance	An activity is strongly favored and its dominance demonstrated in practice
9	Extreme importance	The evidence favoring one activity over another is of the highest possible order of affirmation
2, 4, 6, 8	Intermediate values between the two adjacent judgments	When compromise is needed
Reciprocals	If activity i has one of the above numbers assigned to it when compared with activity j, then j has the reciprocal value when compared with i	
Rationals	Ratios arising from the scale	If consistency were to be forced by obtaining n numerical values to span the matrix

Table 3.2: *Fundamental scale for pairwise comparison [3]*

We created a questionnaire and shared it with experts in the context of the application, who filled it according to criteria from table 3.2, see further details in appendix A. In our case, the expert is person who has experience working with air quality monitoring systems, and has managed air quality systems' data to make decisions. Nevertheless, the mechanism of the questionnaire and the extraction of DQ priorities is very flexible, and allows for taking into account the opinion from people with different backgrounds or profiles. A real example of the questionnaire filled by an expert in air quality monitoring leads to the results in tables 3.3 and 3.5. Note that decimals are used to display the quantities instead of fractions. Also note that the diagonal is always 1, meaning that a dimension compared to itself is equally important. The values over the main diagonal are actually those given by the user preferences, while the values below the main diagonal are reciprocals from those above it, i.e. given the $m \times m$ matrix A, and its elements a_{ij}, where m is the number of dimensions, it satisfies that:

$$a_{ij} \times a_{ji} = 1, \forall\ i, j \tag{3.6}$$

The next step comprises normalizing the matrix in such a way that the sum of every column is equal to one. This is needed to comply with the constrains given before:

$$\bar{a_{ij}} = \frac{a_{ij}}{\sum_{l=1}^{m} a_{lj}} \tag{3.7}$$

The result of this normalization is shown in tables 3.4 and 3.6. These tables also show, in the

last column of each one, the results of the calculated weights for each dimension. The weights are calculated as the mean of each row in the normalized matrix:

$$w_i = \frac{\sum_{l=1}^{m} \bar{a_{il}}}{m} \tag{3.8}$$

Dimension	Accuracy	Precision	Confidence	Completeness	Timeliness	Data Volume	Data Redundancy	Concordance
Accuracy	1.0	5.0	5.0	3.0	1.0	0.3	5.0	1.0
Precision	0.2	1.0	0.2	0.3	1.0	0.3	5.0	1.0
Confidence	0.2	5.0	1.0	1.0	1.0	3.0	7.0	1.0
Completeness	0.3	3.0	1.0	1.0	0.3	1.0	3.0	1.0
Timeliness	1.0	1.0	1.0	3.0	1.0	1.0	5.0	0.2
Data Volume	3.0	3.0	0.3	1.0	1.0	1.0	7.0	1.0
Data Redundancy	0.2	0.2	0.1	0.3	0.2	0.1	1.0	0.2
Concordance	1.0	1.0	1.0	1.0	5.0	1.0	5.0	1.0

Table 3.3: *Pairwise comparison matrix without normalization, for data related dimensions*

Dimension	Accuracy	Precision	Confidence	Completeness	Timeliness	Data Volume	Data Redundancy	Concordance	Weights (w)
Accuracy	0.14	0.26	0.52	0.28	0.09	0.04	0.13	0.16	0.20
Precision	0.03	0.05	0.02	0.03	0.09	0.04	0.13	0.16	0.07
Confidence	0.03	0.26	0.10	0.09	0.09	0.38	0.18	0.16	0.16
Completeness	0.05	0.16	0.10	0.09	0.03	0.13	0.08	0.16	0.10
Timeliness	0.14	0.05	0.10	0.28	0.09	0.13	0.13	0.03	0.12
Data Volume	0.43	0.16	0.03	0.09	0.09	0.13	0.18	0.16	0.16
Data Redundancy	0.03	0.01	0.01	0.03	0.02	0.02	0.03	0.03	0.02
Concordance	0.14	0.05	0.10	0.09	0.47	0.13	0.13	0.16	0.16
Sum of Column	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 3.4: *Pairwise comparison matrix normalized. for data related dimensions*

Suppose that the user is more interested in the DQ evaluated from the data itself instead of some system factors, then μ can be set to 0.9, which leads to the final equation 3.9 for the overall DQ index calculation:

$$\begin{aligned} DQI = 0.9 \cdot (0.20DQ_{accu} + 0.07DQ_{prec} + 0.16DQ_{conf} + 0.10DQ_{comp} + 0.12DQ_{time} \\ +0.16DQ_{volu} + 0.02DQ_{redu} + 0.16DQ_{conc}) \\ +0.1 \cdot (0.12DQ_{util} + 0.16DQ_{acce} + 0.28DQ_{inte} + 0.12DQ_{trus} + 0.20DQ_{arti} + 0.12DQ_{acce}) \end{aligned} \tag{3.9}$$

Dimension	Utility	Accessibility	Interpretability	Reputation	Artificiality	Access Security
Utility	1.0	1.0	0.3	1.0	1.0	1.0
Accessibility	1.0	1.0	1.0	3.0	0.3	1.0
Interpretability	3.0	1.0	1.0	1.0	3.0	3.0
Trust	1.0	0.3	1.0	1.0	0.3	1.0
Artificiality	1.0	3.0	0.3	3.0	1.0	1.0
Access Security	1.0	1.0	0.3	1.0	1.0	1.0

Table 3.5: *Pairwise comparison matrix without normalization. for system related dimensions*

Dimension	Utility	Accessibility	Interpretability	Reputation	Artificiality	Access Security	Weights (v)
Utility	0.13	0.14	0.08	0.10	0.15	0.13	0.12
Accessibility	0.13	0.14	0.25	0.30	0.05	0.13	0.16
Interpretability	0.38	0.14	0.25	0.10	0.45	0.38	0.28
Reputation	0.13	0.05	0.25	0.10	0.05	0.13	0.12
Artificiality	0.13	0.41	0.08	0.30	0.15	0.13	0.20
Access Security	0.13	0.14	0.08	0.10	0.15	0.13	0.12
Sum of Column	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 3.6: *Pairwise comparison matrix normalized, for system related dimensions*

3.5 DQ Evaluation

Once the metrics and the model are defined, one can easily compute DQ per dimension as well as the overall DQ index. Furthermore, data coming from nodes are time series per node, which means DQ will be calculated over time for each node. Our proposal is to have different outputs for DQ evaluation results and let user select any of them depending on the desired granularity. Tables 3.7, 3.8 and 3.9 show the column format or headers of the files that will contain the results. Note that *dim#* are the 1 to *n* chosen dimensions. The suffix *time* means that the result granularity is at one-hour time intervals, while the suffix *node* stands for the result granularity per node, the *total* suffix is a single result per dimensions, and the overall DQ index is given at the end. An hourly window for the calculation of DQ is convenient for the selected application since the nodes report data every minute, hence there would be 60 samples every hour. On the other hand, by averaging these samples, we can compare to data from the reference stations that report a measurement every hour.

node	datetime	DQ-dim1-time	DQ-dim2-time	DQ-dimn-time

Table 3.7: *Hourly DQ per dimension and for each node.*

node	DQ-dim1-node	DQ-dim2-node	DQ-dimn-node	DQI-node

Table 3.8: *DQ per dimension and overall result for each node.*

DQ-dim1-total	DQ-dim2-total	DQ-dimn-total	DQI-total

Table 3.9: *Total DQ per dimension and overall results.*

As a summary, the DQ evaluation strategy encompasses the study of the application to identify the set of dimensions in the context of air quality monitoring and then match the metrics to be used for the single dimension's DQ estimation. Through the PCM the set of weights for the model are found and used to calculate the overall DQ index. Finally, the strategy allows to provide three output results for the presentation of system's DQ at different granularities. To concluded, this approach is simple, but consist on logical steps, and fulfills the goal of using both objective and subjective aspects of DQ to provide a single index that tells about the Overall status of the system's DQ, and allows to see how each dimension is contributing to that status.

Chapter 4

Implementation

This chapter presents the details about our implementation of the software platform to evaluate the DQ of the air quality monitoring application. The software implements the metrics and the model of the strategy proposed in this research work (see chapter 3), allowing the user to visualize the results through a web report.

4.1 DQ Software Design

The design process for implementing the software starts by defining the use cases of the platform, according to the requirements of a user that needs to evaluate DQ. Figure 4.1 depicts the use case diagram of the system, and it is composed by the following use cases:

1. Configure Evaluation Parameters: A setup module where the user should input parameters that are necessary for the DQ evaluation. Details are given in section 4.2.
2. Load Data: A load module to read the datasets specified by the user and transform them into dataframes to be processed by the software. A data cleaning module was added to remove "known" outliers. Details are given in section 4.3.
3. Evaluate Data Quality: A DQ evaluation module that will perform all the DQ assessment over each dimension. Details are given in section 4.4.
4. Upload Result Files: An API to upload result files to a spreadsheet, for the user to easily find and read the DQ status. Details are given in section 4.5.1.
5. Visualize Results: A visualization module to report the DQ indexes and DQ evolution over time. Details are given in section 4.5.2.

As displayed in the use case diagram, the user will have three interactions with the tool, first for loading the data, second for setting up the parameters, and finally for visualizing the report. The architecture of the system is depicted in figure 4.2. In the following subsections, a description for each module is presented, including the algorithms or pieces of code 1 to 9. The Python code is quite readable and, in most of the cases, it was put without modifications, however, some lines were trimmed to show only the important parameters. The full code was uploaded to GitHub. The files are *Total DQ Measurement.ipynb* and *DQ2.py*. Also, an user manual is provided in appendix C.

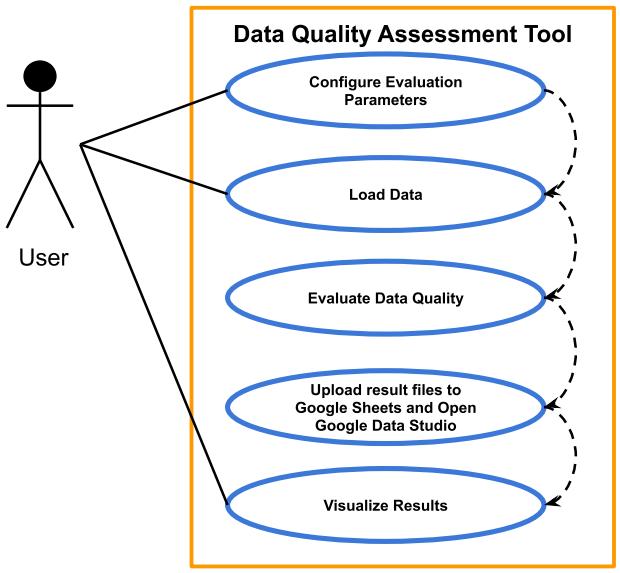

Figure 4.1: *Use Case Diagram.*

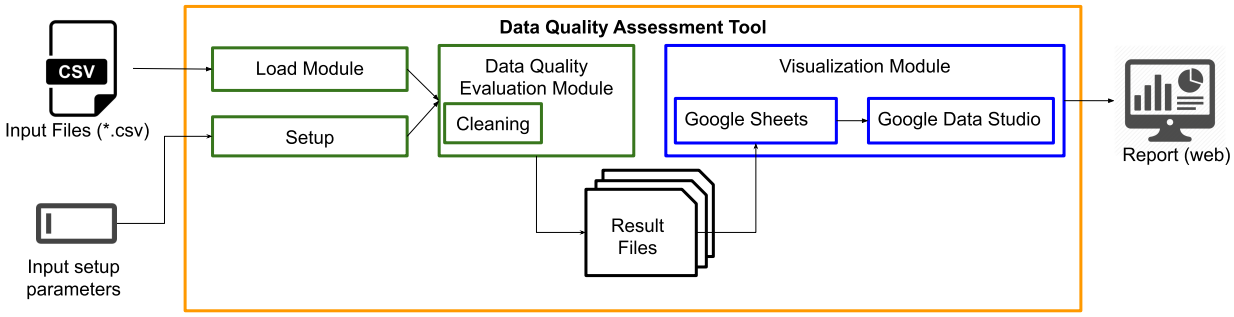

Figure 4.2: *System Architecture.*

4.2 Setup

```
#1. Install the required module packages.
!pip install matplotlib
!pip install scipy
!pip install sklearn
!pip install pandas
!pip install seaborn
!pip install haversine
!pip install -U wxPython
!pip install google
!pip install google-api-core
!pip install --upgrade google-api-python-client
!pip install google-cloud
!pip install google-cloud-vision
!pip install google.cloud.bigquery
!pip install google.cloud.storage
!pip install google-auth-oauthlib
```

Algorithm 1: *Install Packages*

The setup module is responsible for the initialization of the environment and the variables of global parameters used for the DQ assessment. The portion of code 1 refers to the Python packages that were installed on the machine. After the installation, the required modules were imported as displayed in the portion of code 2. We developed the ***DQ2*** module, which implements the main functions for data cleaning and data quality evaluation.

```
#1. Import the required modules.
import multiprocessing as mp
import matplotlib.pyplot as plt
import numpy as np
from scipy import signal
from scipy.stats import uniform
from scipy.stats import norm
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import plot_confusion_matrix
import csv
import pandas as pd
import time
import os
from datetime import datetime, timedelta
import seaborn as sn
import requests
import json
import haversine as hs
import wx
import webbrowser
from __future__ import print_function
import os.path
from googleapiclient.discovery import build
from google_auth_oauthlib.flow import InstalledAppFlow
from google.auth.transport.requests import Request
from google.oauth2.credentials import Credentials
import win32api
import DQ2# Own defined
```

Algorithm 2: *Import Module Packages*

The piece of code 3 presents the setup process. In this step, the user should define the weights, the start and end times, and the confidence level to be used in the consistency DQ evaluation. Note that the weights were previously obtained from the Pairwise Comparison matrix result, as described in appendix A. As stated earlier, these weights reflect the importance that an user gives to each dimension. If wished so, one could freely set these parameters, just guaranteeing that their sum is 1. The start and end times should be within the minimum and maximum timestamps of the datasets. The confidence level is expressed as a percentage, and it should be set to any value desired by the user.

```
#1. Setup weights from Pairwise Comparison results
mu               = 1.0
Waccuracy        = 0.3506311521
Wconfidence      = 0.1880884436
Wconcordance     = 0.1768628272
Wcompleteness    = 0.148093351
Wprecision       = 0.09875987987
Wdata_Redundancy = 0.03756434625

#2. Setup the start_time and end_time of the period to be analyzed:
start_time ="2019-12-01 00:00:00"
end_time   ="2019-12-31 23:59:00"

#3. Define the confidence level p:
p=99.0
```

Algorithm 3: *Setup parameters*

4.3 Load Module

The load module pops up a select file dialog box, implemented using the *get_path()* function, asking the user for choosing 3 files:

- Input the file with the dataset to evaluate (in csv format).
- Input the file with the reference values or ground truth (in csv format).
- Input the file mapping the test nodes to reference stations (in csv format).

Once the files are selected, we read them as Pandas dataframes for their treatment in Python. Appendix C presents details for the csv formats required by our application.

On the line 11 of the piece of code 4, we call the *clean_sort_data()* function, which returns a clean dataset free of inconsistent data, i.e., records out of the sensor ranges and data off $Q3 + 1.5 * IQR$, (outliers). We present more details of this function in table 4.1.

```
#1. Load, as a dataframe, the csv file of the dataset to be evaluated:
df_CC=pd.read_csv(path_for_CC_data)

#2. Load, as a dataframe, the csv file of the reference dataset:
df_SS = pd.read_csv(path_for_SS_data)

#3. Load, as a dataframe, the csv file mapping the test nodes to reference
    stations:
Distances = pd.read_csv(path_for_distance_files)

#4. Use the clean_sort_data(df_CC, df_SS) function to clean and sort the datasets
    :
CC, SS=DQ2.clean_sort_data(df_CC, df_SS)
```

Algorithm 4: *Load Module*

Name	Description	Inputs	Outputs
clean_sort_data()	This function takes the whole dataframes df_CC, df_SS, clean them, sort them by date-time, split them by serial code of the node or station, and returns dictionaries of datasets where the keys are the serial codes and the values are their corresponding dataframes	(df_CC, df_SS)	(CC{codigoSerial:Dataframe}, SS{codigoSerial:Dataframe})

Table 4.1: *clean_ sort_ data() Function*

Besides loading and cleaning the data, we print a summary of the size of the dataset to be analyzed, where it is filtered by the start time and the end time. It shows the number of nodes and the amount of one-minute measurements within the defined period. Similarly, it shows the amount of reference stations and the amount of one-hour measurements within the defined period.

4.4 Data Quality Evaluation Module

The DQ evaluation module is composed of the algorithms 5 and 6. The first one uses the Python *multiprocessing* module for parallelization. The *pool* object is created with the number of available cores in the machine, then the processes to calculate the DQ with the function *eval_ dq()* run independently for every input of the function. For instance, the DQ assessment of every node in the Citizen Science dataset is processed by a different processor in a distributed manner.

The *eval_ dq()* function uses separated sub-functions for the DQ calculation of each dimension, as explained in table 4.2. The second algorithm computes the weighted average. Note that the DQ assessments of each dimension are not separated by DF and NOVA variables, instead an average of both is used. In this study we show our approach only with the PM2.5 variable measured by the DF and NOVA sensors, while the PM10 measurements are ignored, and the relative humidity and the temperature are only used when measuring the correlation, i.e. we do not take into

Name	Description	Inputs	Outputs
eval_dq()	This function takes a list with the node for which the DQ will be calculated, the CC and SS dictionaries, the Distance dataframe, the start and end times and the p confidence level, to evaluate the DQ over all the defined dimensions. It splits the dataset into one-hour groups and runs a *for* loop to evaluate the DQ for every hour. Each dimension has its own function. The results are returned in two dataframes containing the DQ evaluation at one-hour interval periods and the DQ evaluation per node, respectively.	([node, CC, SS, Distances, start_time, end_time, p])	[dim_time, dim_node]
accuracy()	This function calculates the accuracy of a node at one-hour interval periods for the window dataframe. The window contains only data of the specified node and within the defined hour. Other inputs are the distances mapping dataframe and the SS dictionary to get the reference station data. The output is a one-record dictionary that contains the accuracy results of the DF and NOVA sensors in one hour.	(node, hour, window, Distances, SS)	accu_dict_time {accur_df, accur_nova}
precision()	This function calculates the precision of the input window dataframe. The output is a 1-record dictionary that contains the precision results of the DF and NOVA sensors in one hour.	(window)	prec_dict_time {prec_df, prec_nova}
completeness()	This function uses the start and end times to create a reference date-time dataframe and checks whether the window dataframe misses any of its records. This information is used to calculate the completeness. The output is a one-record dictionary that contains the completeness results of the DF and NOVA sensors in one hour.	(node, window, start_time, end_time)	comp_dict_time {comp_df, comp_nova}
duplicates()	This function compares the amount of unique date-time records in the window dataframe to the total number of records, to calculate the rate of repeated data. The output is a one-record dictionary that contains the duplicates DQ result in one hour.	(window)	dupli_dict_time {duplic}
confidence()	This function takes the p confidence level to calculate the confidence of the DF and NOVA measurements in the window dataframe. The output is a one-record dictionary that contains the confidence results of the DF and NOVA sensors in one hour.	(window, p)	confi_dict_time {confi_df, confi_nova}
concordance()	This function takes the same inputs as the accuracy() function and uses them to calculate the correlation between several variables. The output is a one-record dictionary that contains the concordance results of DF and NOVA measurements against temperature and humidity in one hour. Note that the concordance to the SIATA robust station measurements are calculated outside this function and within the eval_dq() function, to compare one-day periods because one-hour periods are not possible.	(node, hour, window, Distances, SS)	conco_dict {concordance_df_nova, concordance_df_siata, concordance_df_hum, concordance_df_temp, concordance_nova_siata, concordance_nova_hum, concordance_nova_temp, }
uncertainty()	This function compares the DF and NOVA measurements using the between sampler/instrument uncertainty as described in [59], to estimate the error among both variables. The output is a one-record dictionary that contains the uncertainty DQ result in one hour.	(window)	uncer_dict_time {uncert}

Table 4.2: *Functions in the DQ2 module*

account their accuracy. Also, it should be noted that to align the node data to the robust stations data, we part from the idea that a robust stations report the accumulated PM2.5 measurements from the last hour, for example, the accumulated measurements from 8:00 AM to 8:59 AM are reported at 9:00 AM. However, the data from the nodes are available every minute, hence, to get a single measurement, we averaged the last hour data, for example, the data reported at 9:00 AM is the average of the measurements taken between 8:00 AM and 8:59 AM (60 samples if data is complete).

```
#1. Start timer t0:
t0= time.time()
print ("Start time: ",t0)

#2. Initialize the dataframes dim_time, dim_node and dim_DQ.

#3. Setup the multiprocessing pool object class with the available CPU number:
pool=mp.Pool(processes = mp.cpu_count())

#4. Setup the per-node input argument_list of the eval_dq(argument_list) function
    , map then to the pool object:
results=pool.map(DQ2.eval_dq,([[nodes, CC, SS, Distances, start_time, end_time, p
    ] for nodes in CC.keys()]))

#5. Extract the DQ evaluation over time dim_time and over node dim_node from the
    results output:
for i in range(0,len(results)):
dim_time=dim_time.append(results[i][0])
dim_node=dim_node.append(results[i][1])

#6. Get the average DQ evaluation from the dim_node dataframe, using the desired
    columns:
dim_DQ= dim_node[cols].mean()

#7. Calculate the elapsed time t1,
t1 = time.time() - t0
print ("Elapsed Time: ", t1)
```

Algorithm 5: *DQ evaluation Modules with Multiprocessing*

```
#1. Add a new column (DQ_INDEX_TOTAL) to the dim_DQ dataframe and fill it the
    weighted average overall calculation of the DQ index:
dim_DQ["DQ_INDEX_TOTAL"]=
      Wprecision*dim_DQ[["precision_average"]]
    + Waccuracy*dim_DQ[["accuracy_average"]]
    + Wcompleteness*dim_DQ[["completeness_average"]]
    + Wconfidence*dim_DQ[["confi_average"]]
    + Wconcordance*dim_DQ[["concordance_average"]]
    + Wdata_Redundancy*dim_DQ[["duplicates_average"]]
```

Algorithm 6: *Weighted average to get the overall DQ Index*

4.5 Visualization Module

The visualization module is composed by two parts, the first one is the Google Sheets API, which is used to export the output dataframes dim_time, dim_node and dim_DQ (whose content and format was described in section 3.5) to their corresponding sheets in the spreadsheet. The second part is the Google Data Studio report, where the data from the spreadsheet is imported and displayed as tables, maps, histograms, and time series.

4.5.1 Google Sheets API

This API was coded using documentation and examples provided by Google. The piece of code 7 shows the main commands to setup the API. The variable SCOPES is where the read/write access is provided, the SPREADSHEET_ID is easily obtained from the spreadsheet url, the credendials.json file is generated in the Google Developers console, the file should be downloaded and saved in the local directory. The spreadsheets.values.clear() method is used to clear sheets before updating the data in them with the spreadsheets.values.update() method. In both methods, it is necessary to pass the SPREADSHEET_ID and the range (sheet_name!cell_range) where the data will be cleared/updated.

```
#1. Define the SCOPES (read/write access):
SCOPES = ['https://www.googleapis.com/auth/spreadsheets']

#2. Provide the SHEETS ID
SPREADSHEET_ID = '1QlPuLYvWaJV6QmOTmkUM3BzuiCvM_8mnuAtvLiEFJaI'

#3. Generate the OAuth 2.0 credentials in the Google Developers console, save the
    file to the local directory:
credentials.json

#4. Build the service:
service=build('sheets', 'v4', credentials=creds)

#5. Call the Sheets API
sheet = service.spreadsheets()

#6. Clear the sheets:
sheet.values().clear(SPREADSHEET_ID,range='DQ_TIME!A1:Z1000000')
sheet.values().clear(SPREADSHEET_ID,range='DQ_NODE!A1:Z1000000')
sheet.values().clear(SPREADSHEET_ID,range='DQ_TOTAL!A1:Z1000000')

#7. Export dim_time, dim_node and dim_DQ dataframes to the designated sheets:
sheet.values().update(SPREADSHEET_ID,range='DQ_TIME!A1',dim_time)
sheet.values().update(SPREADSHEET_ID,range='DQ_NODE!A1',dim_node)
sheet.values().update(SPREADSHEET_ID,range='DQ_TOTAL!A1',dim_DQ)
```

Algorithm 7: *Google Sheets API*

The line of code presented in 8 is a one-click quick way to open the spreadsheet result.

```
#1. Open the Google spreadsheet with the results:
webbrowser.open('https://docs.google.com/spreadsheets/d/1
    QlPuLYvWaJV6QmOTmkUM3BzuiCvM_8mnuAtvLiEFJaI/edit?usp=sharing')
```

Algorithm 8: *To Open the Google spreadsheet*

4.5.2 Google Data Studio Report

Figure 4.3 shows the main interface of Google Data Studio. This tool is used to convert data to graphic reports, and it was preferred over others like Tableau or Power BI because it is easy to integrate to Google Sheets, it is easy to share, it is free, and it has the features required for this project. The designed report has 8 pages, the first one shows data DQ results for all the dimensions, while the others present the results per dimension. It contains a scorecard to show the Overall DQ Index, radar charts to display and compare the total DQ per dimension, time series to show the evolution of DQ in an hourly basis, tables to show the DQ per node for DF and NOVA variables, an interactive map to show the node's locations and their DQ, and histograms to show the DQ distribution of the one-hour records.

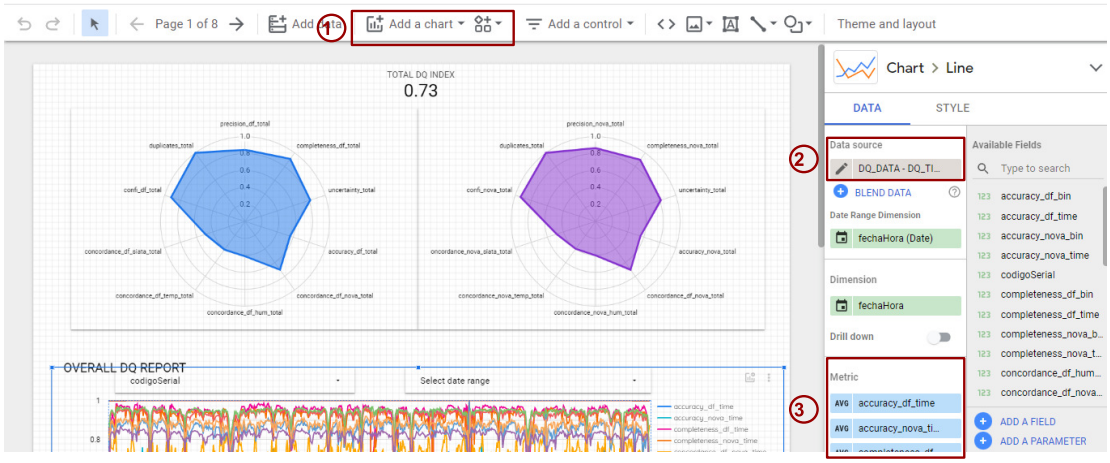

Figure 4.3: *Google Data Studio Interface: 1) Select the Chart, 2) Select the source of Data, 3) From the available fields, select the metrics to be plotted in the chart.*

The line of code presented in 9 is a one-click quick way to open the web report.

```
#1. Open the Google Data Studio web report:
webbrowser.open('https://datastudio.google.com/s/hy-ZhY6eEfU')
```

Algorithm 9: *To Open the Google Data Studio Report*

From this chapter, it can he highlighted the use of software like JupyterLab to program in Python using notebooks, which allows a better segmentation and documentation of code. Github as a repository for version control and backups. Google Sheets, which provides an API that can be accessed from Python, and that allows saving the results online. And Google Data Studio, to

build the online graphical report out of the data quality results in Google Sheets. At the same time, the relationship between each module is clearly established and separated in different cells.

The implementation was meant to comply with the proposed strategy, finding it in the Data Quality Evaluation Module. Furthermore, the tool has other modules that complement the final architecture, and that help the user to interact with it, besides some simple features that look to improve the experience.

Chapter 5

Tests and Results

This chapter exposes the tests carried out and the results of this project. To test the system, three different tasks were proposed. In the first task, the usage of multiprocessing was verified to decrease the processing time in the DQ evaluation module, which is an advantage to process large datasets; it can be checked in section 5.1. Before continuing to the second task, in 5.2 we present the results of the Pairwise Comparison Matrix. In the second task, we evaluated the DQ awareness of the system by using a controlled synthetic dataset created by modifying some parameters on it to check whether the tool was capable to show any variation in DQ, section 5.3 presents the results of this experimentation. In the third task, we evaluate our DQ tool on a real dataset and the results were displayed by the visualization module in a web report, the detailed results are given in section 5.4. Finally, section 5.5 mentions the publications related to this research, where further and specific results are found.

The datasets used in this project are detailed in tables 5.1 and 5.2. The synthetic dataset was built as explained in appendix B. On the other hand, the real dataset corresponds to the application previously described in section 3.1, provided by SIATA.

Name	**Variables of Interest**	**Analyzed Period**	**Number of Nodes**	**Sampling Period**	**Total Records**
SIATA robust Stations	PM2.5	$2021-10-05\ 00:00:00$ $2021-10-08\ 00:00:00$	1	1H	73
Citizen Science	PM2.5, Humidity, Temperature	$2021-10-05\ 00:00:00$ $2021-10-08\ 00:00:00$	10	1Min	43210

Table 5.1: *Synthetic dataset details*

Name	**Variables of Interest**	**Analyzed Period**	**Number of Nodes**	**Sampling Period**	**Total Records**
SIATA robust Stations	PM2.5	$2019-12-01\ 00:00:00$ $2019-12-31\ 23:00:00$	20	1H	14880
Citizen Science	PM2.5 Humidity Temperature	$2019-12-01\ 00:00:00$ $2019-12-31\ 23:59:00$	219	1Min	7524875

Table 5.2: *Real dataset details*

The following results were obtained using the JupyterLab Python IDE in a machine with the following specifications (table 5.3):

Feature	Specification
Processor	AMD Ryzen 5 4500U with Radeon Graphics 2.38 GHz
Installed RAM	8.00 GB (7.42 GB usable)
System type	64-bit operating system, x64-based processor
OS	Windows 10 21H1

Table 5.3: *Machine specifications*

5.1 Time Performance Evaluation

When this tool was first conceived to evaluate DQ in study [13], the running time was measured in hours, however, after further optimization of the code, we could reduce it to minutes. In the process, we figured out that the calculation of the DQ for a single node did not depend on the calculation of other nodes, however, all the code was run in a single thread because of Python Global Interpreter Lock-GIL. Under the premise that the DQ assessment of each node was independent, we decided to use parallelization with the multiprocessing package of Python, which allows to use subprocesses instead of threads by bypassing the GIL. With such implementation, the processing time was further reduced.

For this test, the *processes* argument within the *Pool*(*processes*) function of the multiprocessing module of Python was varied from 1 to 6 (6 was the maximum number of available processors in the machine where the test was performed). And for each CPU number, the test was run 5 times, registering the elapsed time for each run, as shown in tables 5.4 and 5.5. Each table shows the elapsed time taken by the DQ evaluation module to give a result for both the synthetic and the real dataset.

Test Number	Processing Time					
	1 CPU	2 CPUs	3 CPUs	4 CPUs	5 CPUs	6 CPUs
Execution time of test 1 (s)	8.30	5.95	4.93	4.50	4.18	4.64
Execution time of test 2 (s)	8.60	5.97	4.99	4.45	4.44	4.64
Execution time of test 3 (s)	8.35	6.31	4.96	4.43	4.42	4.76
Execution time of test 4 (s)	8.48	5.94	5.22	4.41	4.83	4.66
Execution time of test 5 (s)	8.47	6.21	4.95	4.31	4.57	4.70
Mean time/Seconds	8.44	6.07	5.01	4.42	4.49	4.68
Mean time/Minutes	0.14	0.10	0.08	0.07	0.07	0.08

Table 5.4: *Synthetic dataset processing times vs number of used CPUs*

Figure 5.1 depicts the average time necessary to process both datasets as a function of the number of CPUs. It is evidenced for the real dataset that the processing time was reduced almost by four when the number of processors changed from 1 to 6. However, for the synthetic dataset, the processing time reached a minimum when using 4 processors, but a larger number of CPUs did not decrease the processing time, and it actually seems to increase. This behavior can be explained by the fact that the dataset is too small, and setting up the multiprocessing function can spend more resources than what are actually required by the application, i.e., there is a higher overhead when creating processes over the available CPUs. The benefit of the parallelization are

Test Number	Processing Time					
	1 CPU	2 CPUs	3 CPUs	4 CPU	5 CPUs	6 CPUs
Execution time of test 1 (s)	1336.89	752.55	558.59	450.72	434.76	416.48
Execution time of test 2 (s)	1344.86	758.30	557.29	475.61	429.52	416.50
Execution time of test 3 (s)	1341.02	752.06	550.07	466.66	439.04	416.51
Execution time of test 4 (s)	1336.34	755.67	555.19	452.94	442.59	415.75
Execution time of test 5 (s)	1337.43	750.15	552.52	469.47	424.53	414.77
Mean time/Seconds	1339.31	753.74	554.73	463.08	434.09	416.00
Mean time/Minutes	22.32	12.56	9.25	7.72	7.23	6.93

Table 5.5: *Real dataset processing times vs number of used CPUs*

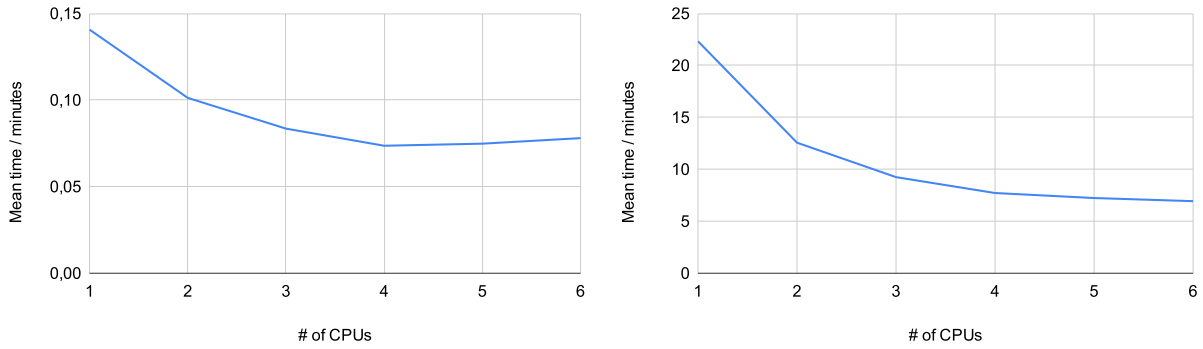

Figure 5.1: *Time Performance vs number of CPUs. Left: Synthetic dataset. Right: Real dataset*

more notorious for large datasets. This is more evident in the speedup graph shown in figure 5.2. This chart shows the ratio between the sequential time (1 CPU) and the parallel time (n CPUs), and stands for the acceleration that is obtained when varying the number of cores. For the synthetic dataset, it stops accelerating with 4 cores, whereas for the real dataset it continues speeding up, and if more cores were available, the processing time could be further reduced.

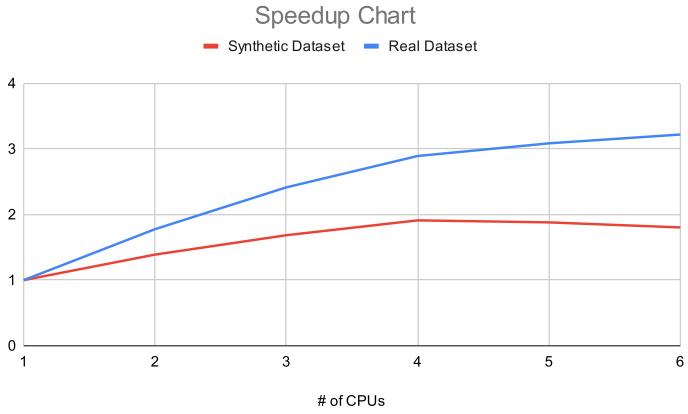

Figure 5.2: *Speedup chart for both the synthetic dataset and the real dataset.*

5.2 Pairwise Comparison Metrics Results

The results presented in this chapter are based on the PCM outcome from the answers given by an expert user who is involved in the field of *Air Quality, Environmental monitoring*. The weights obtained from the answers that other users gave can be compared in table 5.6. **Bold text** was used to mark the maximum and minimum weights. It turns out that the accuracy dimension received the highest weight, while the data duplicates dimension received the lowest weight in 3 out of 4 the cases, indicating that users have preferences for data reflecting the true value, while it is not that important the presence of repeated information. The latter one is actually an easier problem to solve than trying to figure out the true value of a measurement. See appendix A for details about the PCM configuration.

User	Field	Accuracy	Precision	Confidence	Completeness	Duplicates	Concordance	Total
1	Air Quality, Environmental monitoring	**0.35**	0.10	0.19	0.15	**0.04**	0.18	1.00
2	Air Quality, Environmental monitoring	**0.43**	0.22	0.16	0.11	0.06	**0.02**	1.00
3	Sensors, IoT	**0.40**	0.14	0.12	0.07	**0.03**	0.23	1.00
4	Sensors, IoT	**0.35**	0.20	0.14	0.06	**0.02**	0.22	1.00

Table 5.6: *Weights obtained from different user's answers.*

We also computed the Intraclass Correlation Coefficient (ICC) using R, this metric is used to measure the level of agreement of multiple "raters" on several "subjects". The ICC ranges from 0 to 1, where a larger value is desired. The procedure to calculate the ICC was done using the indications in [67], leading to the results displayed in algorithm 10. As the obtained ICC is $0.75 < 0.819 < 0.9$, it can be interpreted as a good reliability or agreement by the different raters. Also, the p-value $1.43e-05 < 0.05$ indicates that the study is significant. These results indicate the effectiveness of the subjective evaluation of DQ that we carried out, that the questionnaire was understood by the users, and that even if some of them are in different DQ fields, they agree on what are the DQ priorities for this application.

5.3 Data Quality Awareness of The Tool

To evaluate the tool in terms of its capability to detect changes of DQ, we created a controlled synthetic dataset which includes some parameters for modifying its behavior in order to induce known changes that would affect DQ. Table 5.7 presents the modifications induced into the dataset, and the results obtained using our tool. The **Test Detail** super column shows the variations manually introduced to the dataset for each **Dimension** in terms of proportions **Prop1** and **Prop2** corresponding to the variables **Var1** and **Var2**, respectively. In this test, we also performed 5 repetitions for each **Dimension**. Later in this section, and for each test, we explain the meaning of the **Prop1** and **Prop2** induced variations.

```
>
> data
                   X1   X2   X3   X4
Accuracy         0.35 0.43 0.40 0.35
Precision        0.10 0.22 0.14 0.20
Confidence       0.19 0.16 0.12 0.14
Completeness     0.15 0.11 0.07 0.06
Data_Redundancy 0.04 0.06 0.03 0.02
Concordance      0.18 0.02 0.23 0.22
> icc(data, model = "twoway", type = "agreement", unit = "single")
 Single Score Intraclass Correlation

   Model: twoway
   Type : agreement

   Subjects = 6
     Raters = 4
   ICC(A,1) = 0.819

 F-Test, H0: r0 = 0 ; H1: r0 > 0
    F(5,15) = 16.1 , p = 1.43e-05

 95%-Confidence Interval for ICC Population Values:
  0.513 < ICC < 0.969
>
```

Algorithm 10: *Intraclass Correlation Coefficient (ICC)*

Next, columns **ACCU**, **PREC**, **COMP**, **DUPL**, **CONF**, **CONCOR**, **UNCER** of table 5.7, show the result DQ levels related to the variation of the DQ for each dimension and for the variables where they were assessed. It might not be intuitive for some dimensions like the data duplicates, that a result close to 1 is an excellent index, for that reason it is important to remember that every DQ index ranges from 0 to 1, where 0 is a poor DQ level, while 1 is an excellent DQ level. The metrics were built in that way to maintain the consistency. The texts in the table were shortened to optimize the space, i.e., **DF** stands for the PM2.5 measurements of the Citizen Science node DF sensor, **NV** stands for the PM2.5 measurements of the Citizen Science node Nova sensor, **ST** stands for the PM2.5 measurements of the robust SIATA station, **HU** stands for the relative humidity measurements of the Citizen Science node sensor, and **TE** stands for the temperature measurements of the Citizen Science node sensor.

Finally, the last column, **DQI**, of table 5.7 presents the results of the overall DQ index as the weighted average of the per-dimension results. To help the reader identify results in the table, **bold text** was used for the table records where changes were evidenced.

		Test Detail				ACCU		PREC		COMP		DUPL	CONF		CONCOR							UNCER	DQI
Dimension	**#**	**Var1**	**Prop1**	**Var2**	**Prop1**	**DF**	**NV**	**DF**	**NV**	**DF**	**NV**	**DF, NV**	**DF**	**NV**	**DF-NV**	**DF-ST**	**DF-HU**	**DF-TE**	**NV-ST**	**NV-HU**	**NV-TE**	**DF, NV**	**Total**
	0	**Synthetic dataset without changes**				0.91	0.91	0.94	0.94	1.00	1.00	1.00	0.98	0.98	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.95
ACCU	1	**DF**	**0.80**			**0.81**	0.91	0.94	0.94	1.00	1.00	1.00	0.98	0.98	1.00	0.94	0.99	0.99	0.94	0.99	0.99	**0.84**	0.93
ACCU	2	**DF**	**0.50**			**0.51**	0.91	0.94	0.94	1.00	1.00	1.00	0.98	0.98	1.00	0.94	0.99	0.99	0.94	0.99	0.99	**0.53**	**0.88**
ACCU	3			**NV**	**1.20**	0.91	**0.77**	0.94	0.94	1.00	1.00	1.00	0.98	0.98	1.00	0.94	0.99	0.99	0.94	0.99	0.99	**0.87**	0.93
ACCU	4			**NV**	**1.50**	0.91	**0.48**	0.94	0.94	1.00	1.00	1.00	0.98	0.98	1.00	0.94	0.99	0.99	0.94	0.99	0.99	**0.72**	**0.88**
ACCU	5	**DF**	**0.80**	**NV**	**1.20**	**0.81**	**0.77**	0.94	0.94	1.00	1.00	1.00	0.98	0.98	1.00	0.94	0.99	0.99	0.94	0.99	0.99	**0.72**	0.91
PREC	1	**DF**	**0.20**			0.91	0.91	**0.79**	0.94	**0.99**	1.00	1.00	**0.93**	**0.98**	**0.29**	**0.94**	**0.29**	**0.28**	0.94	0.99	0.99	**0.86**	0.90
PREC	2	**DF**	**0.50**			**0.89**	0.91	**0.55**	0.94	**0.96**	1.00	1.00	**0.85**	**0.98**	**0.16**	**0.91**	**0.16**	**0.15**	0.94	0.99	0.99	**0.68**	**0.86**
PREC	3			**NV**	**0.30**	0.91	**0.90**	0.94	**0.70**	1.00	**0.99**	1.00	**0.98**	**0.90**	**0.22**	0.94	0.99	0.99	**0.93**	**0.22**	**0.21**	**0.79**	**0.88**
PREC	4			**NV**	**0.40**	0.91	**0.90**	0.94	**0.61**	1.00	**0.98**	1.00	**0.98**	**0.87**	**0.18**	0.94	0.99	0.99	**0.92**	**0.18**	**0.18**	**0.73**	**0.87**
PREC	5	**DF**	**0.20**	**NV**	**0.30**	0.91	**0.90**	**0.79**	**0.70**	**0.99**	**0.99**	1.00	**0.93**	**0.90**	**0.15**	**0.94**	**0.29**	**0.28**	**0.93**	**0.22**	**0.21**	**0.75**	**0.86**
COMP	1	**DF**	**0.10**	**NV**	**0.10**	0.91	0.91	0.94	0.94	**0.90**	**0.90**	1.00	**0.98**	**0.98**	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.94
COMP	2	**DF**	**0.20**	**NV**	**0.20**	0.91	0.91	0.94	0.94	**0.80**	**0.80**	1.00	**0.98**	**0.98**	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.92
COMP	3	**DF**	**0.30**	**NV**	**0.30**	0.91	0.91	0.94	0.94	**0.70**	**0.70**	1.00	**0.97**	**0.97**	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.91
COMP	4	**DF**	**0.40**	**NV**	**0.40**	0.91	0.91	0.94	0.94	**0.60**	**0.60**	1.00	**0.97**	**0.97**	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	**0.89**
COMP	5	**DF**	**0.50**	**NV**	**0.50**	0.91	0.91	0.94	0.94	**0.50**	**0.50**	1.00	**0.97**	**0.97**	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	**0.88**
DUPL	1	**DF**	**0.10**	**NV**	**0.10**	0.91	0.91	0.94	0.94	1.00	1.00	**0.91**	0.98	0.98	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.95
DUPL	2	**DF**	**0.20**	**NV**	**0.20**	0.91	0.91	0.94	0.94	1.00	1.00	**0.83**	0.98	0.98	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.95
DUPL	3	**DF**	**0.30**	**NV**	**0.30**	0.91	0.91	0.94	0.94	1.00	1.00	**0.77**	0.98	0.98	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.94
DUPL	4	**DF**	**0.40**	**NV**	**0.40**	0.91	0.91	0.94	0.94	1.00	1.00	**0.72**	0.98	0.98	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.94
DUPL	5	**DF**	**0.50**	**NV**	**0.50**	0.91	0.91	0.94	0.94	1.00	1.00	**0.67**	0.98	0.98	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.94
CONF	1	**p**	**90.0**	std=0, Comp=1		0.91	0.91	0.94	0.94	1.00	1.00	1.00	**0.99**	**0.99**	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.95
CONF	2	**p**	**95.0**	std=0, Comp=1		0.91	0.91	0.94	0.94	1.00	1.00	1.00	**0.98**	**0.98**	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.95
CONF	3	**p**	**97.0**	std=0, Comp=1		0.91	0.91	0.94	0.94	1.00	1.00	1.00	**0.98**	**0.98**	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.95
CONF	4	**p**	**99.0**	std=0, Comp=1		0.91	0.91	0.94	0.94	1.00	1.00	1.00	**0.98**	**0.98**	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.95
CONF	5	**p**	**99.9**	std=0, Comp=1		0.91	0.91	0.94	0.94	1.00	1.00	1.00	**0.97**	**0.97**	1.00	0.94	0.99	0.99	0.94	0.99	0.99	1.00	0.95
CONCOR		**Tests based on other dimensions**				**Results based on other dimensions**																	
UNCER		**Tests based on other dimensions**				**Results based on other dimensions**																	

Table 5.7: *Summary of Tests and Results of the Tool's DQ Awareness*

Now we discuss each result in detail:

- **Test 0**: These are the raw results of the evaluation on the synthetic dataset changing no parameter on it. Refer to appendix B for details about how the dataset was created and how the modifications were done.

  - It is normal that the accuracy is not 1 because the "true value" corresponds to the hourly measurement from the robust stations. However, the "measured value" corresponds to the hourly average of the n one-minute samples ($n = 60$ if data is complete) within the hour, taken from the low-cost sensors. It means that by default there is a bias when calculating the accuracy. Adding or subtracting an offset to the DF or NOVA variables will make the accuracy decrease.
  - Similarly, it is expected that the precision is not 1 because it stands for dispersion (standard deviation) of the n measurements within the one-hour periods. For example, adding random noise to the DF or NOVA measurements will increase the the dispersion and the precision will decrease.
  - Regarding the completeness, it is 1 since the synthetic dataset does not have missing data. Removing rows from the dataset, i.e. creating missing values, will make the completeness decrease.
  - The data duplicates is 1 since the synthetic dataset does not contain repeated values. Adding repeated records (rows with the same timestamp) will result in a reduction the data duplicates index.
  - Confidence depends on variables like the standard deviation of the sample (in the 1-hour intervals), the number of samples n and the desired confidence level p, avoiding it to reach 1. The confidence level was set to $p = 99\%$ for this test. A variation of any of these three parameters will make the confidence decrease.
  - The concordance of the raw synthetic dataset is 1 or close to 1 because the DF and NOVA measurements were set to the same values. Also, both the DF and NOVA measurements have a high correlation to temperature and humidity because of the regression model constructed. It shows that there is a high linear dependency between the two variables. Changes in the trends or behavior of the variables will make the concordance decrease.
  - Finally, the uncertainty is a DQ indicator added to assess the **error** due to differences between the DF and the NOVA measurements. Its metric (see equation 3.2) turns to be 1 because the between-sampler uncertainty error (see equation 3.1) is 0 , meaning that the DF and NOVA measurements are totally the same. Any variation on one measurement, will make the uncertainty decrease.
  - The overall DQ index for this test is 0.95, evidencing the high quality of the dataset, but the dimension that affected it the most was the accuracy, whose weight is 0.35, followed by confidence with 0.19, concordance with 0.18, completeness with 0.15, precision with 0.09 and data duplicates with 0.04, see table 5.6. The uncertainty was not included in the overall calculation of the DQ index because it is a DQ indicator, moreover, other

measurements of error like accuracy and precision were already included. Note that there are several DQ values for each dimension; for example, the accuracy, the precision and the confidence dimensions have a value for both DF and NOVA measurements. To get a single value per dimension, we averaged the DQ results of DF and NOVA on each dimension. Because of the weight obtained for the accuracy, small changes on it will highly impact the overall DQI, but changes in precision and data duplicates will barely impact the overall DQI.

- **Accuracy Tests 1-5**:
  - In the synthetic dataset, we multiplied the DF PM2.5 variable measurements by 0.8 and 0.5, meaning a reduction of 20% and 50% regarding the true value, respectively. The tool was capable of measuring a reduction of the DF accuracy to 0.81 and 0.51, corresponding to the induced changes.
  - Similarly, the NOVA variable measurements were multiplied by 1.2 and 1.5, meaning an increment of 20% and 50% regarding the true value, respectively. The tool was capable of measuring a reduction of the NOVA accuracy to 0.77 and 0.48, corresponding to the induced changes.
  - Next, both DF and NOVA variables were multiplied by 0.8 and 1.2, i.e. a change of 20% in both of them, for which the tool measured a reduction on the accuracy DF variable to 0.81 and NOVA variable to 0.77. The results did not perfectly matched the change, but again it needs to be mentioned that the comparison of the one-hour reference data to the n averaged samples in the one-hour interval will result in this kind of differences.
  - As expected, the concordance didn't change since the measurements keep the same trend, however, the uncertainty decreased because the difference between DF and NOVA measurements increased.
  - There was no impact on other dimensions. The overall DQI reached a minimum of 0.88 when the accuracy of either of the variables decreased to the half.

- **Precision Tests 1-5**:
  - To change the precision, we added a random error with mean equal to zero and a standard deviation proportional to the mean, in the variables of the dataset. When the standard deviation of the random error was set to 0.2 and 0.5 times the mean of the DF measurements, the tool could assess a reduction in the precision to 0.79 and 0.55, respectively, going in line with the introduced changes.
  - When the random error added to the NOVA measurements was set to zero mean and the standard deviation to 0.3 and 0.4 times the mean, the tool could measure a reduction of the precision to 0.70 and 0.61, respectively, corresponding to the induced changes.
  - One last test consisted of adding a random error with standard deviation equivalent to 0.2 and 0.3 times the mean, to the DF and NOVA measurements, respectively. In both cases, the tool detected a reduction of the precision to 0.79 and 0.70, both in line with

the induced changes. As evidenced in the results, the changes in the precision are very close to the induced modifications, any differences can be explained by the randomness of the added variations.

- As expected, the induced dispersion also impacted on dimensions like the confidence, the concordance and the uncertainty indicator. The accuracy was barely affected since the mean of the error was set to 0, similarly to the concordance between the DF and the SIATA robust station PM2.5 measurements, or the NOVA and the SIATA PM2.5 measurements. The completeness also was impacted because of the data cleaning process prior to the processing.
- The Overall DQ Index was highly impacted by changes in the dispersion of the measurements, which is explained by the effect that such dispersion has on most of the dimensions. The Overall DQ reached a minimum of 0.86 when the standard deviation of the induced random error was to 0.5 times the value of the DF mean, or to 0.2 times the DF mean and 0.3 times the NOVA mean.

- **Completeness Tests 1-5**:

  - The completeness of the dataset was modified by removing the desired proportion of rows from the dataset, the removal was uniformly distributed over the whole dataset. The induced change was varied from 0.1 to 0.5 times the length of the dataset and the tool detected this change each time.
  - As expected, changes in the completeness also impacted on the confidence. These changes barely impacted on other dimensions because of the 1-hour averages, or because the missing values were full rows (all the variables of a row) instead of single variables.
  - The overall DQ index reached a minimum of 0.88 when half of the records were removed from the dataset.

- **Data Duplicates Tests 1-5**:

  - The creation of repeated data was done uniformly over the whole dataset by duplicating the desired amount of rows proportionally to the length of the dataset. The proportion was varied from 0.1 to 0.5 and the results were in line with these changes. The tool measured that the data duplicates dimension index reduced to 0.91, 0.83, 0.77, 0.72 and 0.67. In fact, the tool is obtaining consistent data, since the metric for the data duplicates dimension was defined as $1 - (repeatedvalues)/(collectedvalues)$, meaning that an introducing 0.5 of repeated data will lead to $1 - (0.5)/(1.5) = 0.67$ reduction of data duplicated index.
  - The repeated data barely impacted on other dimensions. Some changes can be spotted in the confidence results, but they are very small.
  - The overall DQ index did not significantly change since the weight for the data duplicates dimension is only 0.04, the smallest one.

- **Confidence Tests 1-5**:

  - During the precision and completeness tests, it was possible to detect their impact on the confidence dimension. For those tests, a confidence level $p = 99$ was used. For the confidence tests, it will be shown how the selection of the confidence level as 90%, 95%, 97%, 99% or 99.9% changes the confidence when the completeness and the precision are not modified.
  - A lower confidence level means a smaller confidence interval where the true value is in, and a higher value in confidence dimension metric. If the user wants a higher confidence level, the interval will be larger and the confidence dimension index will decrease. These results are not necessarily good or bad, they just reflect the user's choice of the confidence level. Contrary to changes in precision and completeness that contribute to a real reduction of the confidence.
  - Even with a confidence weight of 0.19 (the second highest), the overall DQ index is not impacted by changes in the confidence level, since the confidence variation was too small.

- **Concordance and Uncertainty Tests**:

  - No particular tests were done with the concordance dimension. As stated earlier, it is affected by changes in the variables trends or behavior, as was shown with precision tests. It is worth mentioning that a single value for the concordance was obtained by averaging the concordances of DF-NV, DF-ST and NV-ST. The DF-NV result is highly impacted by the introduced errors, however, DF-ST and NV-ST remain stable, allowing the average to increase. Its impact on the overall DQ index is high because of the 0.18 weight.
  - Regarding the uncertainty DQ indicator, it does not impact on the overall DQ index because it was not part of the PCM and we did not assign a weight to it, meaning that it is not part of the weighted average. The tool's evaluation of uncertainty responds to the introduced offsets and errors of the accuracy and precision tests, indicating a difference (an error) between the DF and NOVA PM2.5 measurements.

5.4 Results On The Real Dataset

In this section we present the results for the DQ evaluation in the real dataset, composed by 219 citizen sciences nodes, 20 reference stations, during the month of December, 2019, see table 5.2. The results can be further checked in the web report. In the following figures, we show the captures of each page in the report, and the interpretation and discussion of the results for each DQ dimension.

Figure 5.3 presents the overall DQ evaluation of the whole dataset. The total DQ index is 0.73, and the dimensions that contribute the most to this drop are the accuracy and the concordance. Remember that the indexes range between 0 a 1, where 0 means a bad assessment while 1 stands for an excellent assessment. Other dimensions are over 0.8 in both the DF and the NOVA measurements. The accuracy is 0.56, a low value for a dimension that captures difference

between the measured value and the true value, however, as explained in [13], most of the nodes are within a range of $2km$ to the closest SIATA robust station, some of them can be up to $7km$ far. As found and discussed in [13], there was not found a correlation between the distance and the accuracy, however this distance will cause a bias in the comparison, and also for a region with a topography as the Aburrá Valley, the measurements can significantly change from one location to another, even more if the height difference between the nodes and the SIATA robust stations are not considered. The concordance between DF or NOVA nodes to SIATA stations is 0.5, which is caused by the same issue related to the distances previously discussed. It can be verified that both node sensors' measurements are highly correlated, around 0.7 of concordance between them. The uncertainty of 0.8 can reaffirm this result, indicating that there is a small error between the measurements. The concordance to variables like the relative humidity and temperature is low, around 0.4%, showing that apparently there is not a linear dependency between PM2.5 measurements and these variables. Further studies on the dependency of such variables need to be done but are not part of the scope of this research. For that reason, the overall DQ index does not take them into consideration.

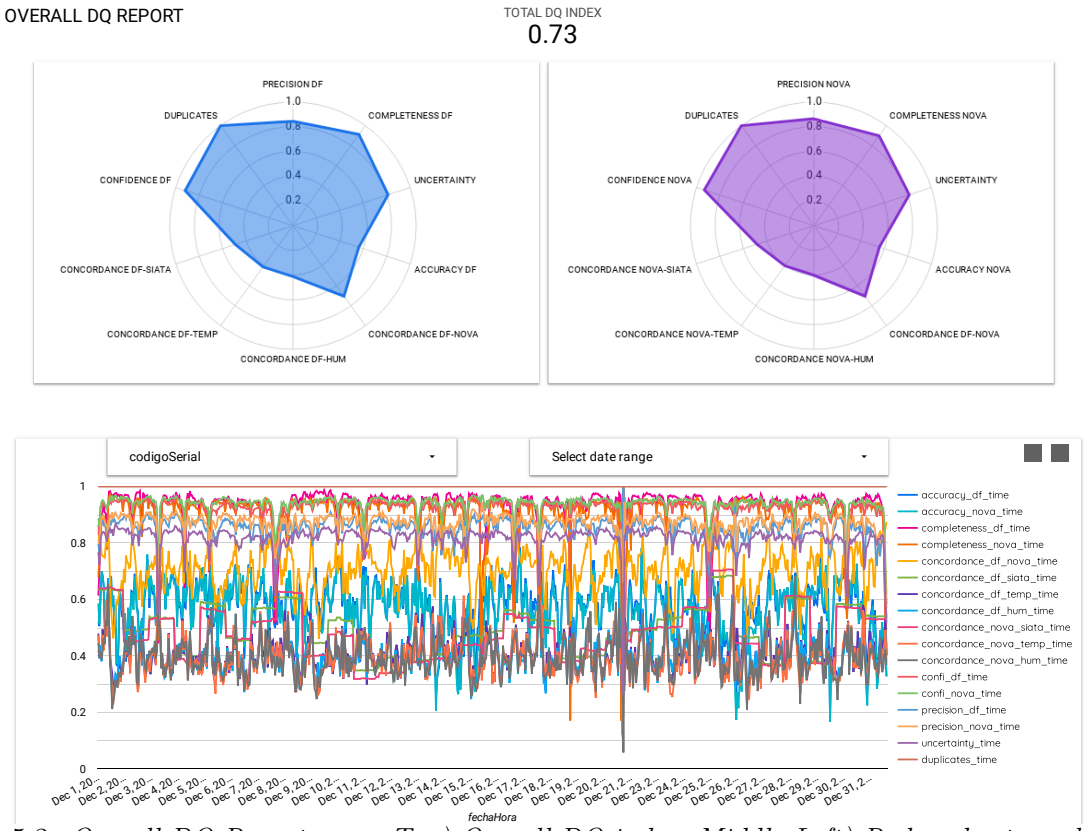

Figure 5.3: *Overall DQ Report page. Top) Overall DQ index, Middle Left) Radar chart results for DF related dimensions, Middle Right) Radar chart results for NOVA related dimensions, Bottom) All dimensions DQ evolution over time at one-hour intervals.*

Detailed information can be obtained from the figure 5.4, where the maps show that the accuracy for DF variable is in the range $[0.04, 0.76]$, while for the NOVA variable it is in the range $[0.00, 0.75]$. Nodes in green color have a higher accuracy, and most of them are in the city, where most of the robust stations are located, i.e. they are close, thus allowing a better comparison

between the measured value and the true value. Also, the histogram shows that 37% of the records (i.e. 1-hour time intervals of the whole dataset) were undefined probably because the closest station did not have data during the same period. Also, 10% of the records have accuracy 0, which means a large difference between the measurement of the nodes and the SIATA stations. The histogram also tells that 10.1% and 9.4% of the records have an accuracy of 0.9, which means that apart from the undefined data and the zero accuracy, most of the remaining records have a high accuracy. In the time series of the figure, it can be seen that the mean (over the nodes and for each hour) accuracy is around 0.6 and remains stable. However, by the end of the month, it seems to degrade.

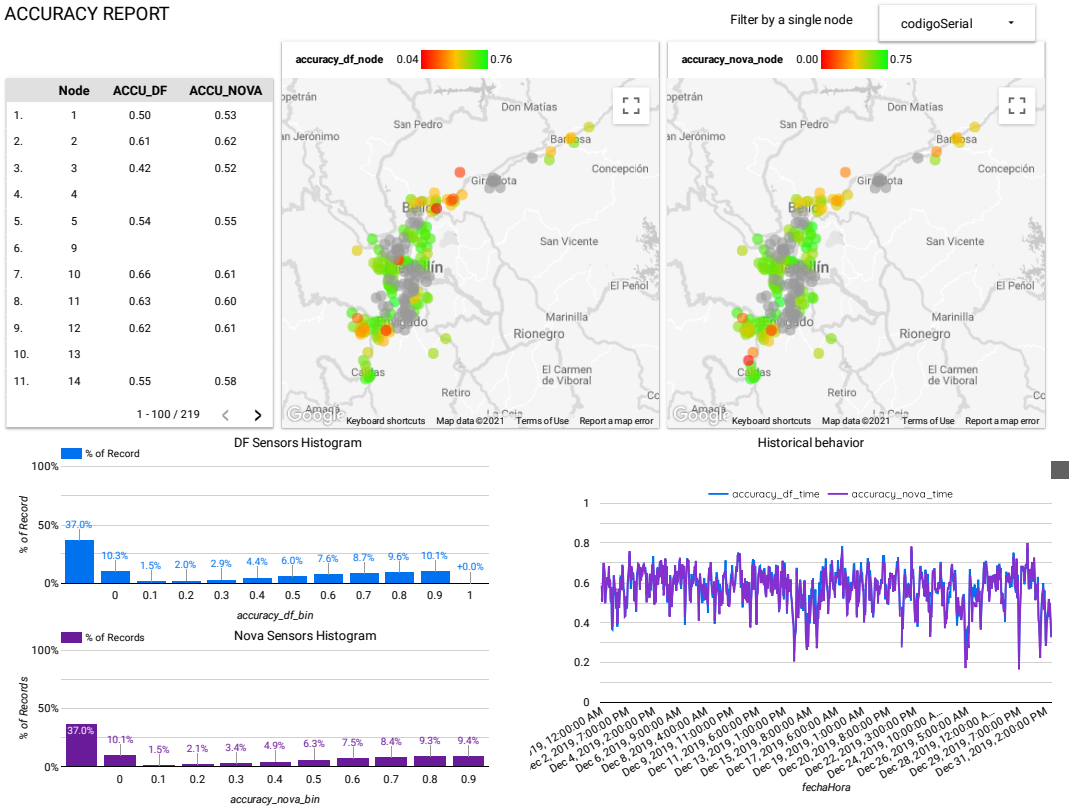

Figure 5.4: *Accuracy Report page.*

Figure 5.5 shows the precision report. The precision ranges from $[0.03, 0.95]$ in DF sensors and $[0.00, 0.98]$ in NOVA sensors. The histogram shows that the dispersion of around half of the records (48.9% and 55.1%) is 0.9, standing for a dispersion less that the 10%, a really good value that tells the PM2.5 concentrations did not vary too much within the one-hour periods. The time series shows that the average precision is near 0.9, being the NOVA sensor more precise. The precision remains stable during the entire month, except at the end where it seems to decrease.

The completeness report in figure 5.6 indicates that the completeness is within the range $[0.00, 0.99]$. The histogram shows that the completeness is greater than or equal to 0.9 for 87% of the DF sensor records and 79% of the NOVA sensor records. In the time series, apart from the frequent drops (which actually appear during daytime, probably caused by sensor saturation

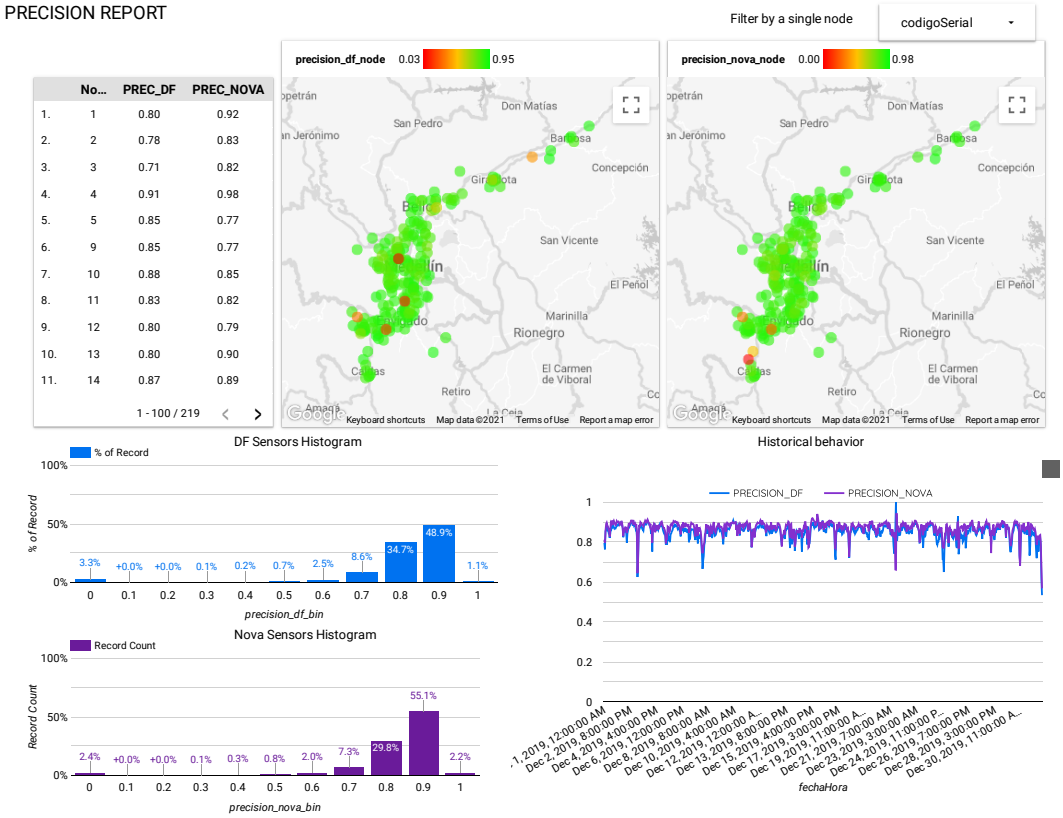

Figure 5.5: *Precision Report page.*

leading to missing data), the trend remains stable around 0.9. As mentioned in [13], missing values can be caused by the initial cleaning process of data out of range, the fact that sensors rely on the user's power grid and internet service, both of which are exposed to outage, sensors out of services, malfunctioning, misuse, lack of maintenance, etc.

Regarding the data duplicates report, as shown in all the components of figure 5.7, all of the records are 1, meaning that there is no presence of repeated data.

The confidence report is displayed in figure 5.8. It shows that the ranges for DF and NOVA data qualities are $[0.03, 0.98]$ and $[0.00, 0.99]$, respectively. According to the histogram, over 91.4% of records have a confidence of 0.9, and the time series depicts a stable trend of the confidence during the month. The results of the confidence depend on the completeness and precision, both of which showed good figures, and the confidence level that was set to 99%. One could think that, with a 99% confidence, the true value of PM2.5 measurements of both DF and NOVA sensors is in a range of ± 0.1 times the mean, however, it is not necessarily so because the accuracy results were not as good as the completeness and precision results. For instance, the confidence analysis should be complemented with the accuracy analysis. Uncertainty and concordance DQ dimensions could also be helpful.

In the second last part is the concordance dimension report, depicted in figure 5.9. It shows that the range of the concordance between the DF and NOVA measurements is $[0.15, 1.00]$. The histogram tells that the correlation of the two variables is strong to very strong (greater than or

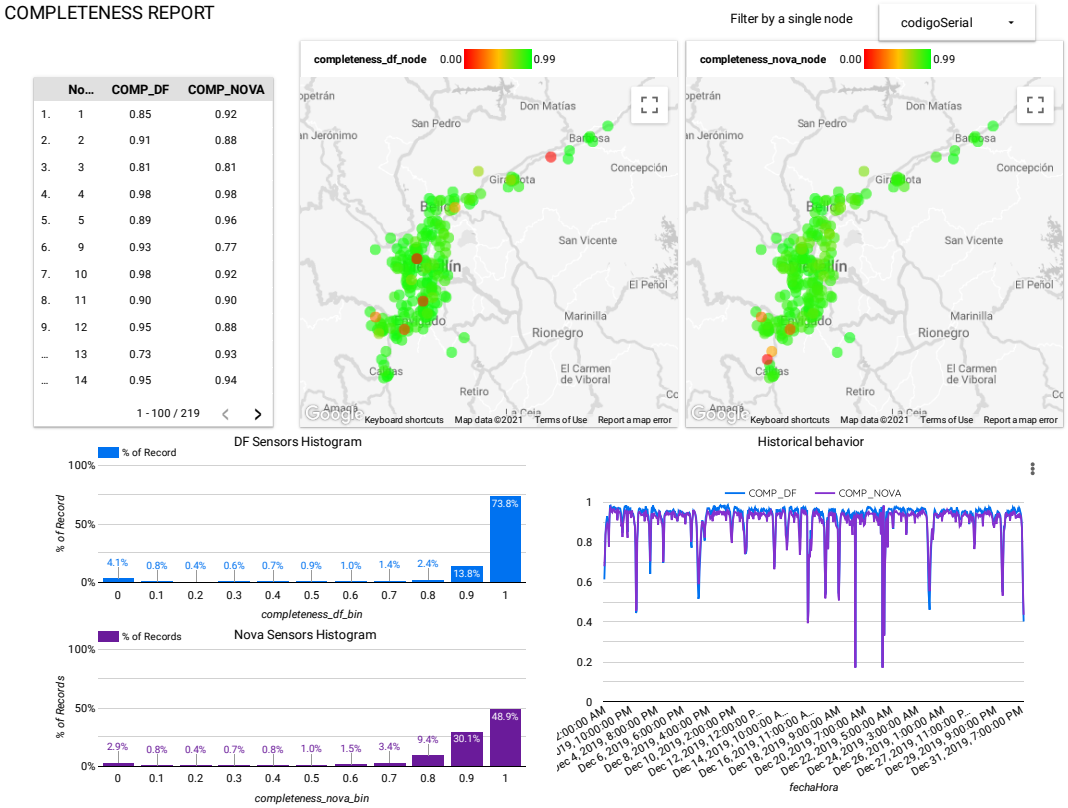

Figure 5.6: *Completeness Report page.*

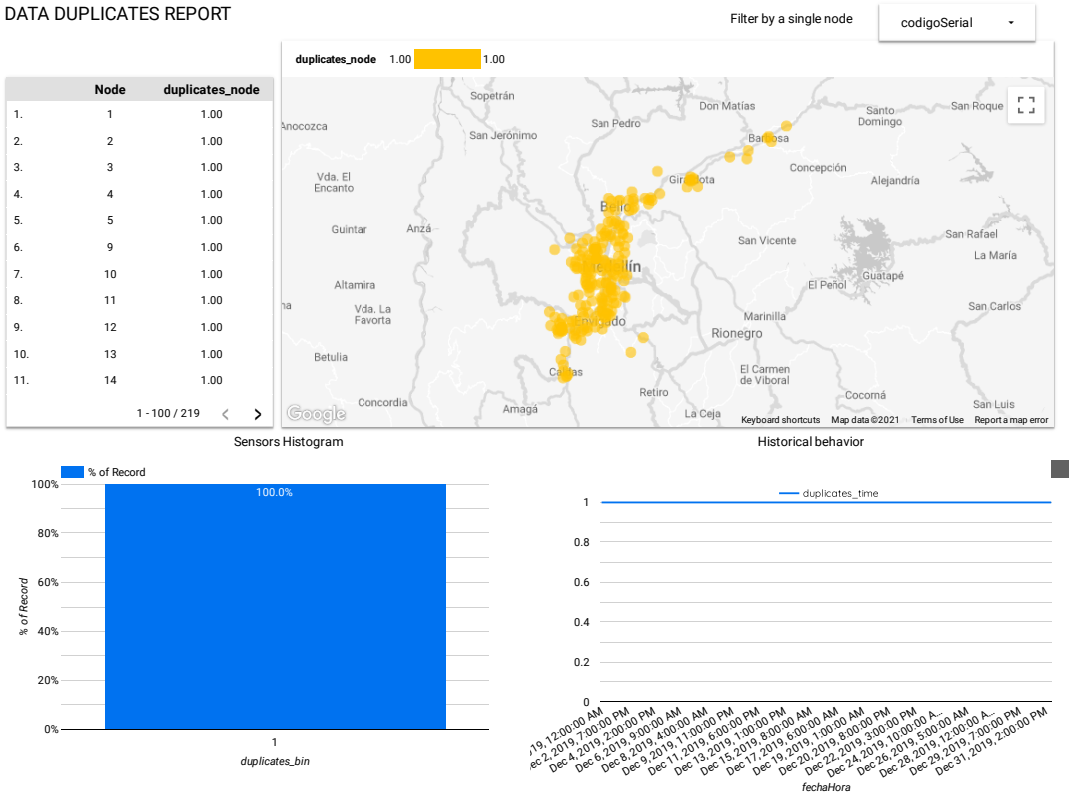

Figure 5.7: *Data Duplicates Report page.*

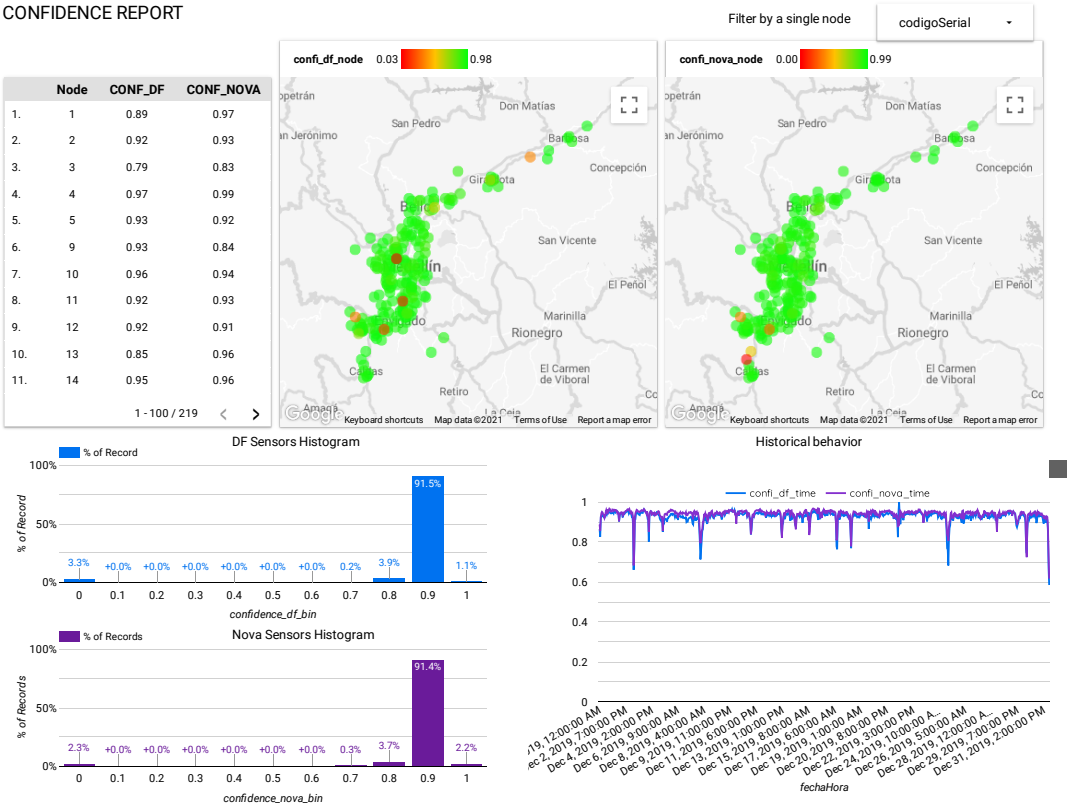

Figure 5.8: *Confidence Report page.*

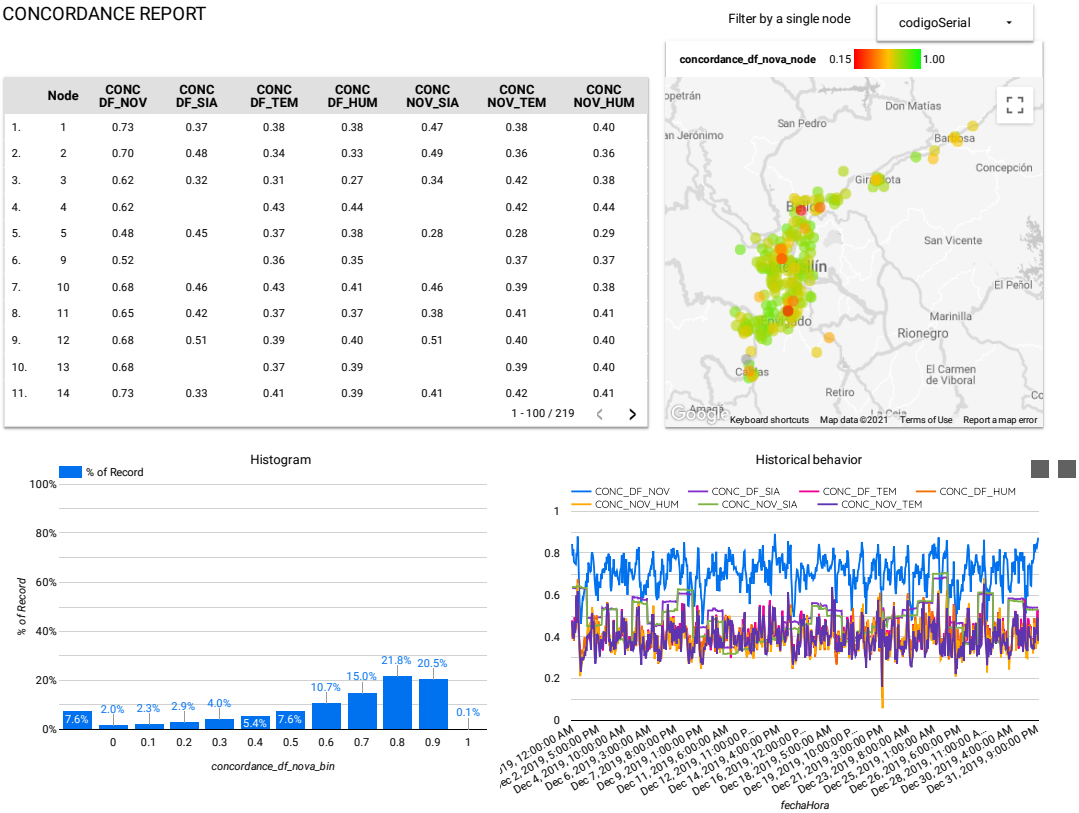

Figure 5.9: *Concordance Report page.*

equal to 0.6) for about 68% of the records. The time series shows the evolution of the correlation of all variables during the month. The concordance to the SIATA measurements is higher, somewhere around 0.5, and concordance to temperature and relative humidity is somewhere around 0.4.

Finally, the uncertainty report is shown in figure 5.10. From this figure, we note that the range of the uncertainty is $[0.01, 0.94]$. In the histogram, we can see that 76.3% of the records have an uncertainty greater than or equal to 0.8, i.e. a 20% error between both measurements, and almost half of the records have an error of 10%. The time series confirms this behavior and also evidences a stable trend during the month. Nodes with a high uncertainty index have that behavior because there is a difference between the measurements of the DF and the Nova sensors. Lack of maintenance, loss of calibration and sensor aging can explain this difference. It needs to be considered that other sources of uncertainty are ignored. This requires further analysis out of the scope of this research.

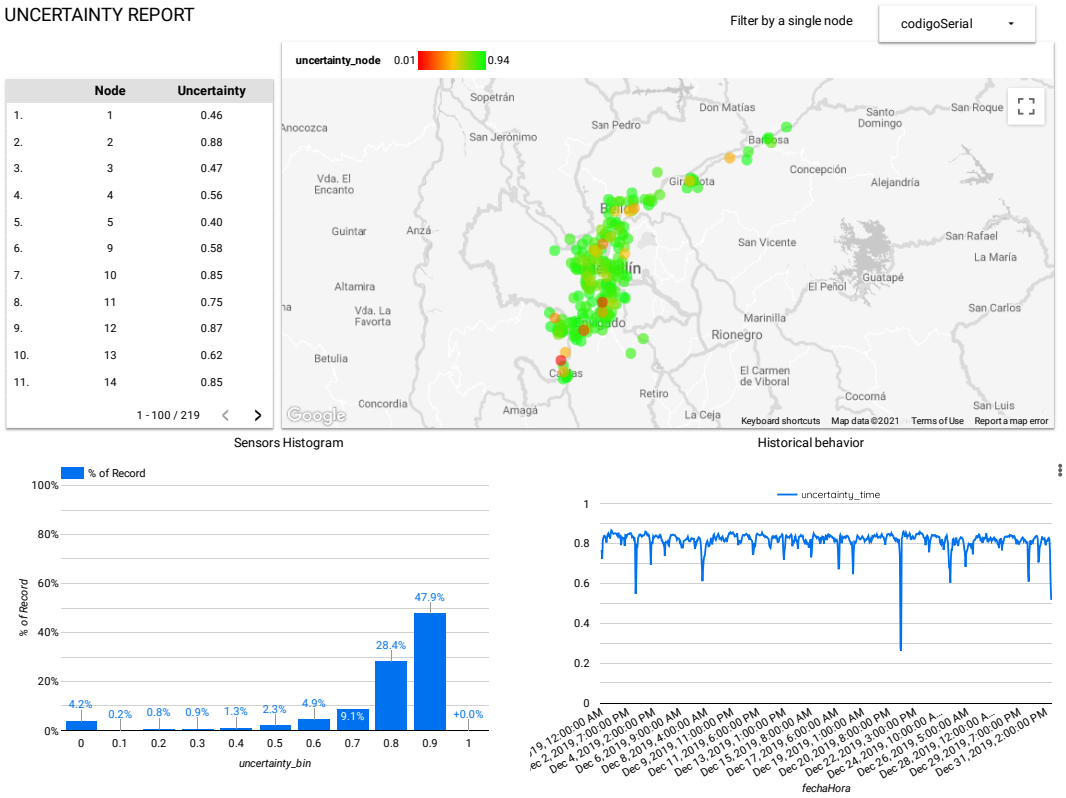

Figure 5.10: *Uncertainty Report page.*

Having showed and discussed the results of the proposed test, we can conclude that using parallelization can improve the time performance of the tool by taking advantage of the cores available in the machine. It would help to perform faster DQ assessments in large datasets, those that are common in IoT applications. Furthermore, we showed that the tool is aware of changes in DQ, and how each dimension affects the overall DQ assessment based on the defined weights. The weights reflect the user's DQ priorities, and correspond to the subjective input for the DQ assessment. We noted a preference for the accuracy dimension, while the data duplicates was

the least preferred dimension. Additionally, the web report contains different graphics that allow to see the behavior of DQ in different ways, may them be location-based, distribution-based or time-based views, which are convenient for the DQ analysis of the system. Finally, the tests and results on the real dataset show the suitability of the tool to learn about the DQ of the system, and to raise conclusions on possible aspects that can be checked to enhance the DQ. To the best of our knowledge, a tool as complete as ours does not exists, and we have shown how this kind of tools can be very useful in the given context.

5.5 Publications

The development of this research lead to the publication of one article [13], where it was presented a first exploration of DQ in the context of air quality monitoring and some preliminary results were given, showing the feasibility of evaluating DQ in terms of single data attributes of an application. In addition, as of October, 2021, a Systematic Mapping Review about DQ in IoT based AQ monitoring networks is in the final writing and edition stage and it will be published soon. Finally, a third publication is being planned to present further details of the strategy, the tool and the results of this research in the field of Data Quality in the Internet of Things.

Conclusions

Among the different data quality fields like definitions, analysis of problems and endangering factors, measurement of data quality, and design of data quality enhancing tools & techniques, a gap related to the data quality assessment was identified. The reviewed articles did not study the data quality on a multidimensional basis or just focused on a few of them. In many cases, the DQ metrics or evaluation techniques were not clear, and the data quality dimensions' names changed from study to study. We did not identified the inclusion of subjective DQ preferences, in spite of they are naturally present in the data quality definition. Finally, it was found that an open challenge was the use and advantages of using a single DQ index to evaluate the DQ of an IoT system.

In this research, we studied the data quality term and its usage in Internet-of-Things-based systems, which lead to the identification and definition of IoT data quality dimensions and their metrics, i.e. it is the way how data quality is approached, not only in the context of IoT but also in IT systems, databases and specific applications like air quality monitoring. We identified, defined and provided with metrics, a total of 15 dimensions in the context of IoT. After narrowing the study to an air quality monitoring system, a set of 11 indicators were also identified, ratifying the concept that each application has its own DQ attributes of interest. In this way, a mapping between IoT and air quality monitoring DQ attributes was proposed, and based on it, the metrics for the air quality application were defined. When analyzing the application, the amount of DQ dimensions was narrowed as well, because in air quality monitoring there is no concern about dimensions like the utility of data, its accessibility, interpretability, artificiality, accessibility, trust and access security. Hence, we focused the study on 6 dimensions, namely accuracy, precision, confidence, concordance, completeness and duplicate, and the uncertainty indicator. Other dimensions, such as timeliness and data volume, were not considered because the characteristics of the application and the dataset did not allow or required it.

After being clear about the application and the dimension set, we proposed to use the Pairwise Comparison Matrix technique for obtaining the user's preferences about DQ dimensions. These preferences reflect the subjective part of the DQ analysis proposed in this research, and gather what are the most important attributes of the DQ product for that user. As expected, we found that the accuracy dimension received the highest weight, while the data redundancy received the lowest one in most of the cases, indicating that the surveyed users had preferences for data reflecting the true value, while the presence of repeated information is not that important. With the dimensions and their weights, a model was proposed and used in a tool coded in Python.

We proposed a Python tool that implements the DQ evaluation model. The tool uses multiprocessing to leverage the analysis of large datasets, it was shown how the processing time was reduced more than 3 times when using the 6 available cores of the machine. In addition, we tested

our implementation over a controlled synthetic dataset, which allowed to compare a clean scenario with all DQ indexes at excellent levels (near 1), to customized scenarios to evidence the induced changes separately by dimension. The results showed that the tool was capable of accurately identify the changes in DQ, per dimension as well as their impact on the overall DQ index, whose sensibility obeyed to the assigned weights.

Our tool allows the user to publish the summarized results in a web report, by using APIs from the tool to Google Sheets and to Google Data Studio. This report is interactive and allows to apply filters to identify the DQ per node and their DQ evolution over time. Based on this information, the user can be informed about the DQ status of the application, can analyze it by single attributes, and can use it to make decisions or not based on the DQ levels. In the same way, our tool can feed applications with this data to automate the process of decision making.

For example, the accuracy in the Citizen Science application was assessed as 0.56, a low value indicating that decisions should not be made based on this information, however, it must be understood that there is a reason for that value and it is because of the distance between the station with "true value" and the node with the measured value. To better assess the accuracy, the true value should be estimated first at the node's location. Other dimensions like the concordance, or even the uncertainty indicator, can complement the information based on which a decision will be made. They can be used to check whether there is or not correlation to other variables and to estimate an error between the measurements of the same variable by two co-located sensors (expecting that two sensors will not be wrong at the same time).

The proposed approach informs about the DQ status and gives insights about the status of the system, allowing to check on specific degraded features and to target improvements on the system's infrastructure to mitigate problems that impact on the retrieved data. For example, bad accuracy could be related to lack of calibration. Bad precision and confidence suggest a high presence of noise. Bad completeness and repeated data mean energy problems, communication problems, or sensor problems. Bad concordance, when a high concordance is expected, means that there is a broken sensor. Low uncertainty means that one sensor is not working properly, etc.

Finally, we consider that the proposed strategy can be used in other applications, and that, as the tool is sufficiently documented, it can be adapted for evaluating other applications and their respective dimensions. A user can define additional dimensions and its metrics in the Python modules, and modify the code for implementing these calculations. Also, the use of free, widely accepted and documented software like Jupyter Notebooks, Github, Google Sheets and Google Data Studio, ease the tasks of sharing and customization of the tool.

Besides what we proposed and developed in this research, where we showed the feasibility of assessing DQ in terms of its attributes, there are still open challenges that are part of future research directions. We identify two main directions, the first one is based on the results of this research, and the second one is related to enhancing the tool. The assessment results can be used to develop and trigger strategies for sensor maintenance, calibration, selection and isolation, to improve not only the system's performance but its data quality, and provide trustable data to the user or to upper layers of IoT applications. It is also important to measure the improvement that can be achieved in decision making when using this kind of tools for being aware of the system's DQ.

Another research direction is to enhance the accuracy assessment by including a model to estimate the "true value" at the node location. It is also pending to test the tool with other applications that involve more dimensions, like those identified as "system's dimensions". In this work, we considered integrating the Pairwise Comparison Matrix process into the tool, however, it was discarded because it is a process that is done only once and there was not really need for it, however, a more robust tool can include it. Finally, this tool can be provided with an API to connect applications that wish to evaluate their data quality on demand or even at run-time.

Appendices

Appendix A

Questionnaire For Pair-wise Comparison Matrix

Section 3.4.3 described in detail the Pair-Wise Comparison method. In this appendix we present the design of the form and the weights that were obtained from the answers given by an expert user. We provide further results and discussion in chapter 5.

Firstly, an expert user was asked to fill this PCM form, where we provided a definition for each dimension, in addition to the meaning of the fundamental scale scoring described in table 3.2. The questions ask to choose only one answer when comparing one dimension to any other. An example is displayed for the accuracy dimensions in the left side of figure A.1. The process is repeated for other dimensions, the only difference is that the next dimension to compare will not include the comparison to accuracy since it was already done. Another thing to note is that system/context dimensions went through the same process separately. See the right side of figure A.1 for an example.

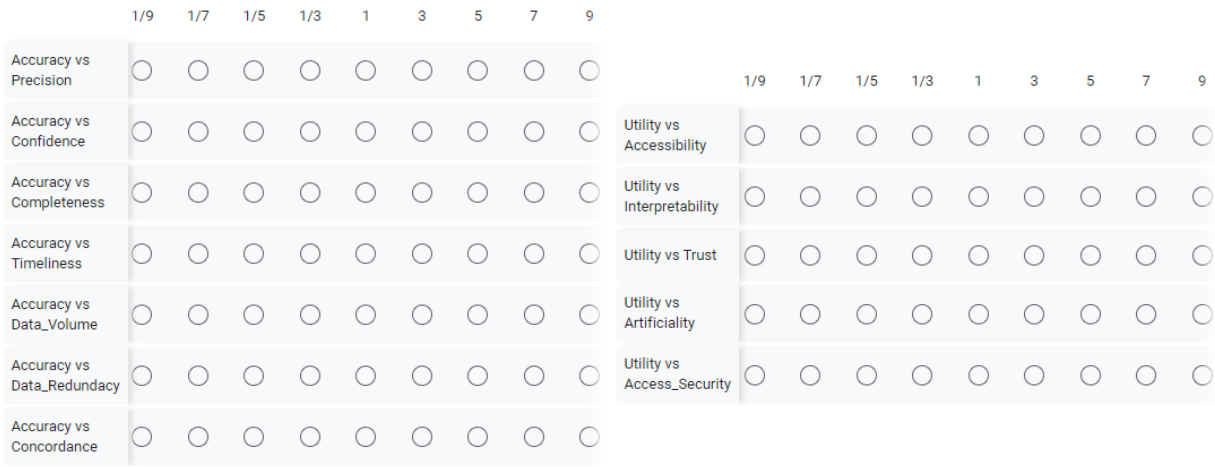

Figure A.1: *Left: PCM between accuracy and other dimensions. Right: PCM between utility and other dimensions.*

The answers were stored in a Google spreadsheet, where further processing described by equations 3.7 and 3.8, was automatically executed, leading to the results in figures A.2 and A.3. In this project, we analyzed the accuracy, precision, confidence, concordance, completeness and duplicates DQ dimensions. For this reason, the timeliness and data volume dimensions had to be

Pairwise Comparison Matrix

		1 Accuracy	2 Precision	3 Confidence	4 Completeness	5 Timeliness	6 Data_Volume	7 Data_Redundancy	8 Concordance
1	Accuracy	1,0	5,0	5,0	3,0	1,0	0,3	5,0	1,0
2	Precision	0,2	1,0	0,2	0,3	1,0	0,3	5,0	1,0
3	Confidence	0,2	5,0	1,0	1,0	1,0	3,0	7,0	1,0
4	Completeness	0,3	3,0	1,0	1,0	0,3	1,0	3,0	1,0
5	Timeliness	1,0	1,0	1,0	3,0	1,0	1,0	5,0	0,2
6	Data_Volume	3,0	3,0	0,3	1,0	1,0	1,0	7,0	1,0
7	Data_Redundancy	0,2	0,2	0,1	0,3	0,2	0,1	1,0	0,2
8	Concordance	1,0	1,0	1,0	1,0	5,0	1,0	5,0	1,0
	Sum of Column	6,9	19,2	9,7	10,7	10,5	7,8	38,0	6,4

Normalization

		1 Accuracy	2 Precision	3 Confidence	4 Completeness	5 Timeliness	6 Data_Volume	7 Data_Redundancy	8 Concordance	Weights
1	Accuracy	0,14	0,26	0,52	0,28	0,09	0,04	0,13	0,16	0,20
2	Precision	0,03	0,05	0,02	0,03	0,09	0,04	0,13	0,16	0,07
3	Confidence	0,03	0,26	0,10	0,09	0,09	0,38	0,18	0,16	0,16
4	Completeness	0,05	0,16	0,10	0,09	0,03	0,13	0,08	0,16	0,10
5	Timeliness	0,14	0,05	0,10	0,28	0,09	0,13	0,13	0,03	0,12
6	Data_Volume	0,43	0,16	0,03	0,09	0,09	0,13	0,18	0,16	0,16
7	Data_Redundancy	0,03	0,01	0,01	0,03	0,02	0,02	0,03	0,03	0,02
8	Concordance	0,14	0,05	0,10	0,09	0,47	0,13	0,13	0,16	0,16
	Sum of Column	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Figure A.2: *PCM results for the first group of dimensions.*

Pairwise Comparison Matrix

		10 Utility	11 Accessibility	12 Interpretability	13 Trust	14 Artificiality	15 Access_Security
10	Utility	1,0	1,0	0,3	1,0	1,0	1,0
11	Accessibility	1,0	1,0	1,0	3,0	0,3	1,0
12	Interpretability	3,0	1,0	1,0	1,0	3,0	3,0
13	Trust	1,0	0,3	1,0	1,0	0,3	1,0
14	Artificiality	1,0	3,0	0,3	3,0	1,0	1,0
15	Access_Security	1,0	1,0	0,3	1,0	1,0	1,0
	Sum of Column	8,0	7,3	4,0	10,0	6,7	8,0

Normalization

		10 Utility	11 Accessibility	12 Interpretability	13 Trust	14 Artificiality	15 Access_Security	Weights
10	Utility	0,13	0,14	0,08	0,10	0,15	0,13	0,12
11	Accessibility	0,13	0,14	0,25	0,30	0,05	0,13	0,16
12	Interpretability	0,38	0,14	0,25	0,10	0,45	0,38	0,28
13	Reputation	0,13	0,05	0,25	0,10	0,05	0,13	0,12
14	Artificiality	0,13	0,41	0,08	0,30	0,15	0,13	0,20
15	Access_Security	0,13	0,14	0,08	0,10	0,15	0,13	0,12
	Sum of Column	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Figure A.3: *PCM results for the second group of dimensions.*

removed from the PCM results. The new results are shown in figure A.4.

The same questionnaire was presented to several users, and the process for calculating the weights was the same in each case. The results for the PCM outcome from other users' answers were presented in table 5.6 of chapter 5.

Pairwise Comparison Matrix

		1 Accuracy	2 Precision	3 Confidence	4 Completeness	5 Timeliness	6 Data_Volume	7 Data_Redundancy	8 Concordance
1	Accuracy	1,0	5,0	5,0	3,0			5,0	1,0
2	Precision	0,2	1,0	0,2	0,3			5,0	1,0
3	Confidence	0,2	5,0	1,0	1,0			7,0	1,0
4	Completeness	0,3	3,0	1,0	1,0			3,0	1,0
5	Timeliness								
6	Data_Volume								
7	Data_Redundancy	0,2	0,2	0,1	0,3			1,0	0,2
8	Concordance	1,0	1,0	1,0	1,0			5,0	1,0
	Sum of Column	2,9	15,2	8,3	6,7	0,0		26,0	5,2

Normalization

		1 Accuracy	2 Precision	3 Confidence	4 Completeness	5 Timeliness	6 Data_Volume	7 Data_Redundancy	8 Concordance	Weights
1	Accuracy	0,34	0,33	0,60	0,45			0,19	0,19	0,35
2	Precision	0,07	0,07	0,02	0,05			0,19	0,19	0,10
3	Confidence	0,07	0,33	0,12	0,15			0,27	0,19	0,19
4	Completeness	0,11	0,20	0,12	0,15			0,12	0,19	0,15
5	Timeliness									
6	Data_Volume									
7	Data_Redundancy	0,07	0,01	0,02	0,05			0,04	0,04	0,04
8	Concordance	0,34	0,07	0,12	0,15			0,19	0,19	0,18
	Sum of Column	1,00	1,00	1,00	1,00			1,00	1,00	1,00

Figure A.4: *PCM results for the first group of dimensions after removing the timeliness and data volume dimensions.*

Appendix B

Synthetic Dataset Generation

The full code for the synthetic dataset generation was uploaded to the GitHub repository as the *ArtificialDataset.ipynb* file. In this appendix we provide the main steps and the overall design of the dataset.

The synthetic dataset was created based on real PM2.5 measurements data from a SIATA robust station, whose data was checked to be complete and error-free. The chosen station was the one with serial code 90, and the period was initially from $01-03-2020\ 00:00:00$ to $04-03-2020\ 00:00:00$, however, the dates were shifted to simulate and up-to-date period: $05-10-2021\ 00:00:00$ - $08-10-2021\ 00:00:00$. The process was like this:

1. An order 2 polynomial interpolation was used to estimate the 1 minute values, increasing the resolution, just like the Citizen Science low-cost sensor nodes. Figure B.1 shows both the Robust Station and the interpolated node PM2.5 behavior in the mentioned period.

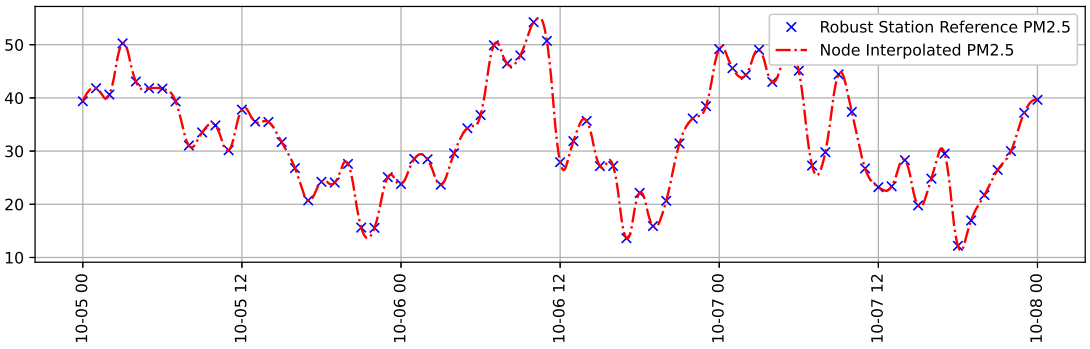

Figure B.1: *Robust Station PM2.5 measurements and interpolated Citizen Science node measurements.*

2. As the temperature and relative humidity variables are also used during the DQ evaluation (concordance dimension), then those variables were generated based on an order 3 polynomial regression model that was created from the PM2.5 measurements of a randomly chosen node. The data corresponds to the node with serial code 67, during the one day period from $02-03-2020\ 00:00:00$ to $03-03-2020\ 00:00:00$. The time behavior of the variables is shown in figure B.2. Note that the pm25_df variable was smoothed with a moving average filter.

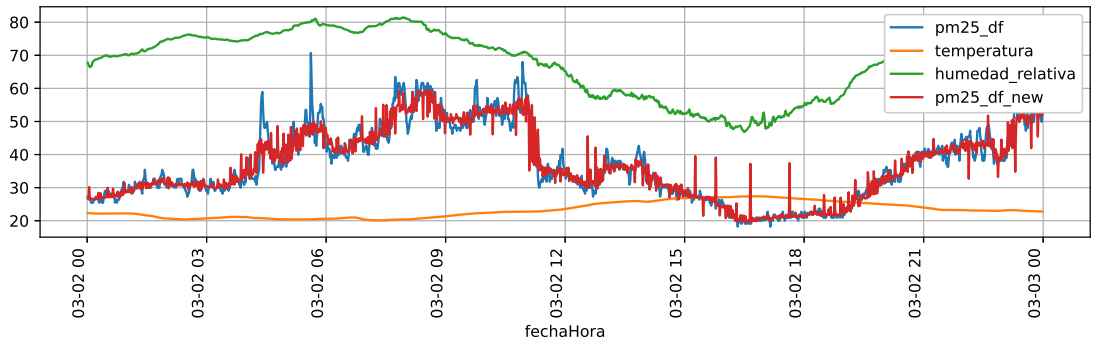

Figure B.2: *Time series of node* 67 *variables.*

$$Temp(PM25) = (-2.15e-04) \cdot PM25^3 + (3.02e-02) \cdot PM25^2 - (1.46e+00) \cdot PM25 + 46.17 \tag{B.1}$$

$$Hum(PM25) = (3.20e-04) \cdot PM25^3 - (6.26e-02) \cdot PM25^2 + (4.00e+00) \cdot PM25 - 7.27 \tag{B.2}$$

The order 3 polynomial models that were obtained for the estimation of temperature and relative humidity based on the PM2.5 pollutant are given in equations B.1 and B.2, respectively. The model fitness was estimated with the R-Squared values 0.46 and 0.61 for the Temperature vs PM2.5 and the Relative vs PM2.5 models, respectively. Those values are not as high as desired and indicate a bad fit of the model, however, this result is enough for the purpose of the synthetic dataset creation. The comparison between the models and the original data is shown in figure B.3.

The idea for these models is to count with a synthetic dataset based on real data, modelling the temperature and relative humidity to have a behavior similar than in real life.

3. The next step is to use the model to estimate the temperature and relative humidity for the node synthetic dataset described in step 1. The final (clean) synthetic dataset has the following variables of interest: $pm25\_df$, $pm25\_nova = pm25\_df$, $temperatura = Temp(pm25\_df)$, and $humedad\_relativa(pm25\_df)$. It is shown in figure B.4.

4. Above process was replicated 10 times to get 10 nodes data.

5. Now that the dataset is ready, the next step was to induce changes on it, which is done to test the tool's DQ awareness. The pieces of code 11, 14, 12 and 13 show how the changes were introduced into the dataset.

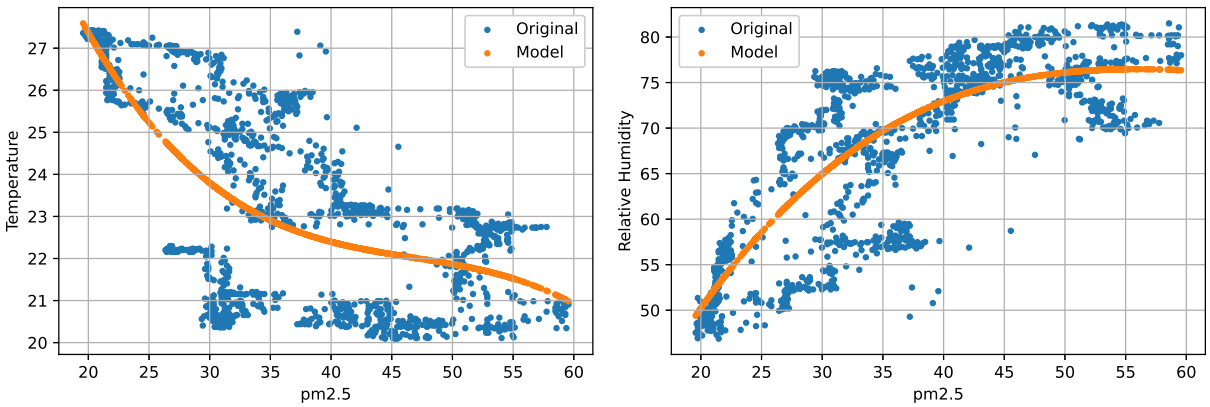

Figure B.3: *Original vs Model comparison*

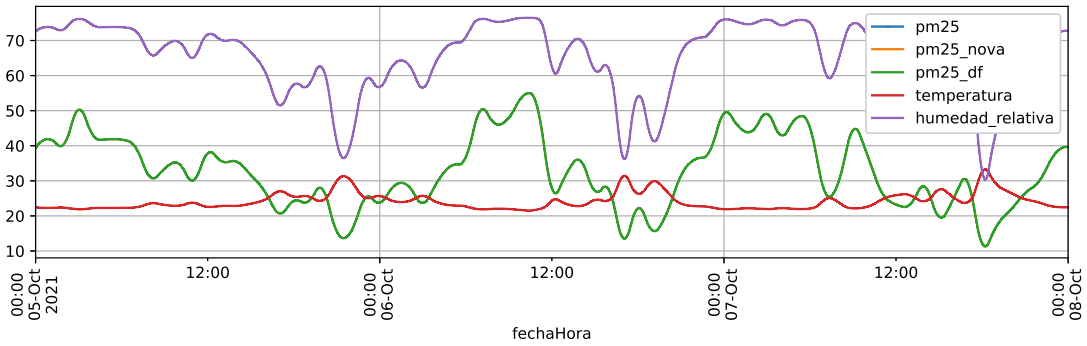

Figure B.4: *Final Synthetic clean dataset for one node, note that PM2.5 measurements are overlapped.*

```
# ACCURACY: add an offset to the the df or nova measurements
Prop1=0.8
Prop2=1.2

CC["pm25_df"]=CC["pm25_df"]*Prop1
CC["pm25_nova"]=CC["pm25_nova"]*Prop2
```

Algorithm 11: *Add and offset to modify the accuracy*

```
# Completeness: randomly remove some data
Prop1=0.7

samples2remove=random.sample(range(len(CC)), round(Prop1*len(CC)))
for i in samples2remove:
    CC.drop(i,axis='index',inplace=True)
```

Algorithm 12: *Remove data to modify the completeness*

```
# Data duplicates: randomly add some data
Prop1=0.5

samples2add=random.sample(range(len(CC)), round(Prop1*len(CC)))
for i in samples2add:
    CC=CC.append(CC.iloc[i], ignore_index = True)
```

Algorithm 13: *Add repeated values to modify the data duplicates*

Figure B.5 shows the effect that the change done for the accuracy using algorithm 11 had on the dataset. Similarly, figure B.6 shows the effect that the change done for the precision (using algorithm 14) had on the dataset. Regarding the completeness, it was difficult to display in the same chart limits, hence it was required to zoom in to appreciate the change, see figure B.7. The algorithm 12 was used in this case.

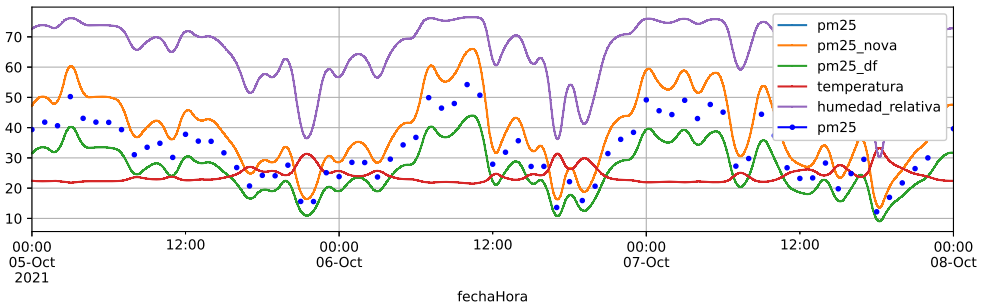

Figure B.5: *Accuracy change.*

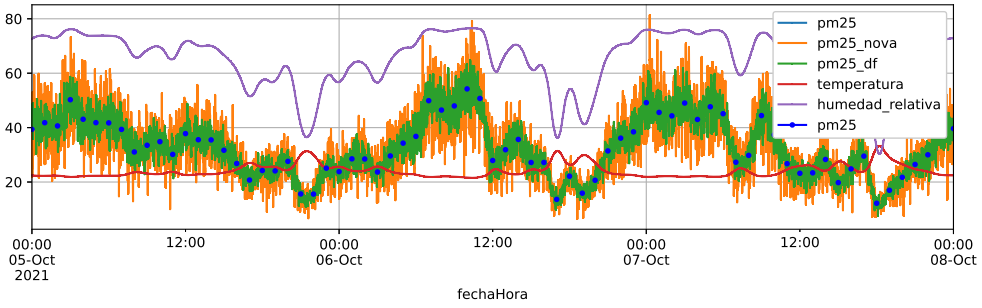

Figure B.6: *Precision change.*

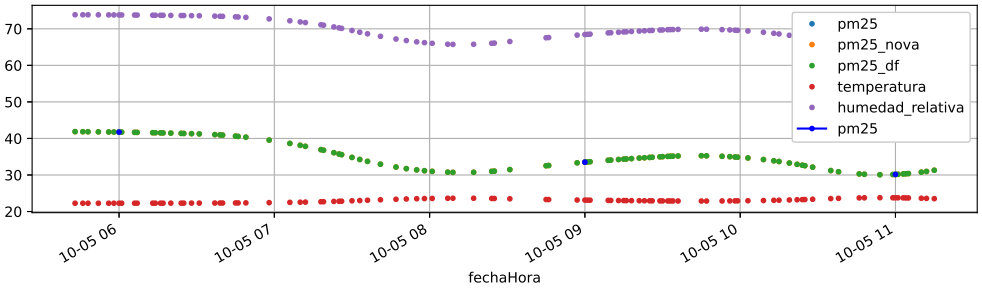

Figure B.7: *Completeness change, it needed to be zoomed in to appreciate the changes.*

Algorithm 13, to create repeated records, was also used but there is no point on plotting it.

```
# PRECISION: add random values.
Prop1=0.2 #for DF Standard deviation added error
Prop2=0.3 #for NOVA Standard deviation added error

hourly_groups=CC.groupby([CC.fechaHora.dt.floor('60min')])
i=0
for hour in hourly_groups.groups.keys():

    np.random.seed(i)
    i+=1
    window=hourly_groups.get_group(hour)
    #print(window)
    mean=window.pm25_df.mean()
    #print(hour, mean)


    indexes=(CC.fechaHora>=hour.floor('60min')) & (CC.fechaHora<(hour+timedelta(
    minutes=1)).ceil('60min'))
    #print(indexes)
    #print(len(CC.loc[indexes,"pm25_df"]))
    CC.loc[indexes,"pm25_df"]=  CC.loc[indexes,"pm25_df"]+  np.random.normal(0,
    Prop1*mean, size=len(CC.loc[indexes,"pm25_df"]))
    np.random.seed(i+1)
    CC.loc[indexes,"pm25_nova"]=CC.loc[indexes,"pm25_nova"]+np.random.normal(0,
    Prop2*mean, size=len(CC.loc[indexes,"pm25_nova"]))
```

Algorithm 14: *Add normal random error to modify the precision*

Regarding other dimensions, such as confidence, concordance and uncertainty, they are also impacted by changing parameters for the accuracy, precision and completeness test, hence they do no need to be plotted.

Appendix C

Tool's User Manual

Navigating in the notebook is straightforward, even more when there is a markdown title before each cell, and the code is documented. However, the code does not run all at once and it is required that the user run every cell. Chapter 4 described the main modules of the tool. In this appendix it is shown the flow diagram of figure C.1, which provides information on what cells are mandatory.

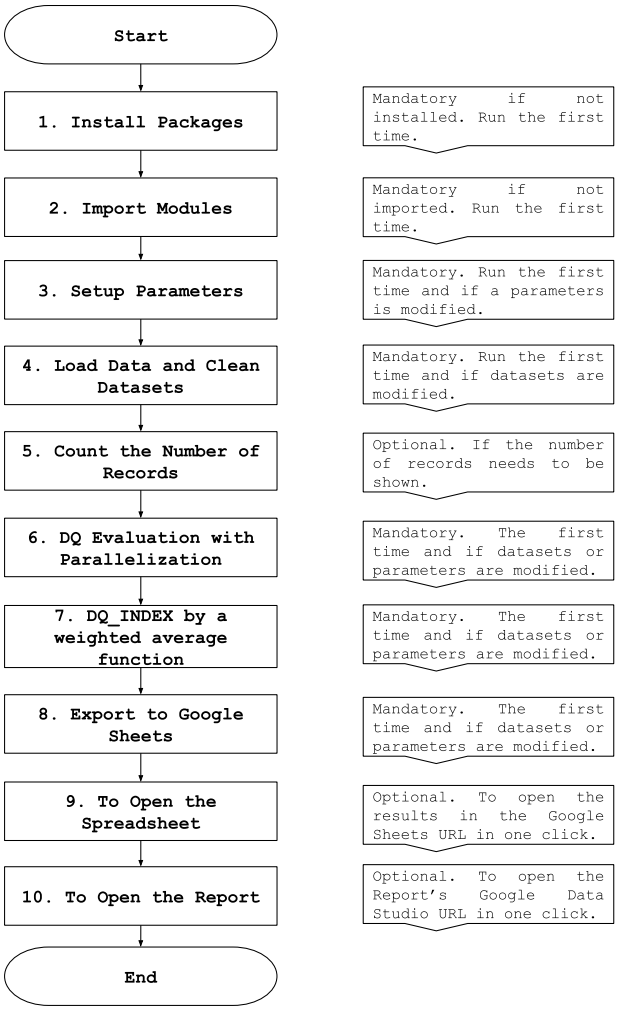

Figure C.1: *Tool User Manual.*

The flow diagram shows the order of execution, what cells are mandatory to be executed and what cells are optional. There are three point of failures related to the user interaction:

- Firstly, in cell 3. *Setup Parameters*, it is needed to make sure that the sum of weights is 1, that the p confidence level is a percentage in the range $[0, 100)$, and that the start and end times are properly set within the range of the datasets to be analyzed, in format "$\%Y - \%m - \%d\ \%H : \%M : \%S$", for example "$2021 - 10 - 05\ 00 : 00 : 00$".

- Secondly, in cell 4. *Load Data and Clean Datasets*, the tool will request to select three files, the order to choose them matters. For instance, the Citizen Science *.csv file should be selected first, then the SIATA robust Stations *.csv file, and finally the Distance mapping *.csv file. If not chosen correctly the tool will show an error in the cell's output.

- Finally, the Citizen Science and the SIATA robust Stations *.csv files, do not have headers, they are added in the Load module, however, the columns are distributed like this:

  - Citizen Science: header_CC=["codigoSerial", "fecha", "hora", "fechaHora", "temperatura", "humedad_relativa", "pm1_df", "pm10_df", "pm25_df", "pm1_nova", "pm10_nova", "pm25_nova", "calidad_temperatura", "calidad_humedad_relativa", "calidad_pm1_df", "calidad_pm10_df", "calidad_pm25_df", "calidad_pm1_nova", "calidad_pm10_nova", "calidad_pm25_nova"]

  - SIATA Stations: header_SS=["Fecha_Hora", "codigoSerial", "pm25", "calidad_pm25", "pm10", "calidad_pm10"]

  - The Distance mapping *.csv file does have header and it only have 4 columns: ["codigoSerial_CC", "codigoSerial_ES", "Distancia_a_ES", "codigoSerial_ES2"]

Acknowledgements

Being away from the classrooms, and considering going back to them, was not easy. On the one hand, I wanted to do some postgraduate studies, while on the other hand, I was in a kind of comfort zone, and also, I felt a little bit rusty about taking classes again, after some years. However, I was very fortunate to count with people on my side that somehow supported me, and inspired me to do it. In the end, it turned out to be challenging, but also a great learning and growing experience.

First, I want to thank my mom for her advice and concern about my education. I also want to thank my sister for cheering me up, my father for what he taught me, my family for always receiving me with open arms when I visited them, my friends for inspiring me to go further, and for their support words when I needed them to start this project, and to keep going. Thanks to my girlfriend for her support and understanding when we couldn't share more time because I was busy. I also want to thank my boss for giving me the chance to study while working. Thanks to my advisors for their patience, their support, their ideas during all the process, and for letting me know, since the beginning, that "we were going to have fun". Thanks to SIATA's team for their kindness and openness to share data about their network. Thanks to my University, teachers, and classmates, for their willingness to teach and their support during the process, and for the knowledge and skills it got from them. Also, thanks to myself for putting the effort into accomplishing this objective, for staying up late, for my willingness to learn and grow. And finally, but not least, I want to thank God for my health and all of the above said.

Medellín, 2022

Julio Hernán Buelvas Pérez

Bibliography

[1] A. Karkouch, H. Mousannif, H. Al Moatassime, and T. Noel, "Data quality in internet of things: A state-of-the-art survey," *Journal of Network and Computer Applications*, vol. 73, pp. 57–81, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2016.08.002

[2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE internet of things journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[3] T. L. Saaty, "How to make a decision: the analytic hierarchy process," *European journal of operational research*, vol. 48, no. 1, pp. 9–26, 1990.

[4] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the internet of things (iot)," *IEEE Internet Initiative*, vol. 1, no. 1, pp. 1–86, 2015.

[5] K. L. Lueth, "State of the iot 2018: Number of iot devices now at 7b – market accelerating," Aug 2018. [Online]. Available: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/

[6] H. B. Sta, "Quality and the efficiency of data in "smart-cities"," *Future Generation Computer Systems*, vol. 74, pp. 409–416, 2017.

[7] L. Berti-Equille, "Measuring and modelling data quality for quality-awareness in data mining," in *Quality measures in data mining*. Springer, 2007, pp. 101–126.

[8] H. A. Khattak, H. Farman, B. Jan, and I. U. Din, "Toward integrating vehicular clouds with iot for smart city services," *IEEE Network*, vol. 33, no. 2, pp. 65–71, 2019.

[9] C. Liu, P. Nitschke, S. P. Williams, and D. Zowghi, "Data quality and the Internet of Things," *Computing*, 2019. [Online]. Available: https://doi.org/10.1007/s00607-019-00746-z

[10] R. Y. Wang and D. M. Strong, "Beyond accuracy: What data quality means to data consumers," *Journal of management information systems*, vol. 12, no. 4, pp. 5–33, 1996.

[11] R. Y. Wang, "A product perspective on total data quality management," *Communications of the ACM*, vol. 41, no. 2, pp. 58–65, 1998.

[12] M. Scannapieco, *Data Quality: Concepts, Methodologies and Techniques. Data-Centric Systems and Applications*. Springer, 2006.

[13] J. H. Buelvas P., F. E. Avila B., N. Gaviria G., and D. A. Munera R., "Data quality estimation in a smart city's air quality monitoring iot application," in *2021 2nd Sustainable Cities Latin America Conference (SCLA)*, 2021, pp. 1–6.

[14] M. Batini, C.; Scannapieca, *Data Quality: Concepts, Methodologies and Techniques*, 2006.

[15] P. Eagan and S. Ventura, "Enhancing value of environmental data: Data lineage reporting," *Journal of Environmental Engineering (United States)*, vol. 119, no. 1, pp. 5–16, 1993, cited By 7. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0027518550&doi=10.1061%2f%28ASCE%290733-9372%281993%29119%3a1%285%29&partnerID=40&md5=dadac7c3a4b78babc3d1592d6e09ecf8

[16] Y. Makoondlall, S. Khaddaj, B. Makoond, and K. Kethan, "Zdlc : Layered lineage report across technologies," 2017, pp. 638–641, cited By 0. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85026652165&doi=10.1109%2fCSE-EUC-DCABES.2016.252&partnerID=40&md5=3ee64453f21b86bd5c10d17d6a1bb635

[17] M. Allen and D. Cervo, "Chapter 9 - data quality management," in *Multi-Domain Master Data Management*, M. Allen and D. Cervo, Eds. Boston: Morgan Kaufmann, 2015, pp. 131–160. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128008355000099

[18] L. Sebastian-Coleman, "Chapter 3 - data management, models, and metadata," in *Measuring Data Quality for Ongoing Improvement*, ser. MK Series on Business Intelligence, L. Sebastian-Coleman, Ed. Boston: Morgan Kaufmann, 2013, pp. 27–37. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123970336000031

[19] M. M. Farooqi, H. A. Khattak, and M. Imran, "Data quality techniques in the internet of things: Random forest regression," in *2018 14th International Conference on Emerging Technologies (ICET)*. IEEE, 2018, pp. 1–4.

[20] F. Li, S. Nastic, and S. Dustdar, "Data quality observation in pervasive environments," in *2012 IEEE 15th International Conference on Computational Science and Engineering*. IEEE, 2012, pp. 602–609.

[21] N. G. Weiskopf and C. Weng, "Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research," *Journal of the American Medical Informatics Association*, vol. 20, no. 1, pp. 144–151, 2013.

[22] N. Javed and T. Wolf, "Automated sensor verification using outlier detection in the internet of things," in *2012 32nd International Conference on Distributed Computing Systems Workshops*. IEEE, 2012, pp. 291–296.

[23] S. Gill, B. Lee, and E. Neto, "Context aware model-based cleaning of data streams," in *2015 26th Irish Signals and Systems Conference (ISSC)*. IEEE, 2015, pp. 1–6.

[24] F. Li, S. Nastic, and S. Dustdar, "Data quality observation in pervasive environments," *Proceedings - 15th IEEE International Conference on Computational Science and Engineering, CSE 2012 and 10th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, EUC 2012*, pp. 602–609, 2012.

[25] G. H. Haan, J. Van Hillegersberg, E. De Jong, and K. Sikkel, "Adoption of wireless sensors in supply chains: a process view analysis of a pharmaceutical cold chain," *Journal of theoretical and applied electronic commerce research*, vol. 8, no. 2, pp. 138–154, 2013.

[26] A. Kothari, V. Boddula, L. Ramaswamy, and N. Abolhassani, "Dqs-cloud: A data quality-aware autonomic cloud for sensor services," in *10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*. IEEE, 2014, pp. 295–303.

[27] J. B. Borges Neto, T. H. Silva, R. M. Assunção, R. A. Mini, and A. A. Loureiro, "Sensing in the collaborative internet of things," *Sensors*, vol. 15, no. 3, pp. 6607–6632, 2015.

[28] A. Dmitriev, E. Efremova, and M. Y. Gerasimov, "Multimedia sensor networks based on ultrawideband chaotic radio pulses," *Journal of Communications Technology and Electronics*, vol. 60, no. 4, pp. 393–401, 2015.

[29] M. Pouryazdan, B. Kantarci, T. Soyata, L. Foschini, and H. Song, "Quantifying user reputation scores, data trustworthiness, and user incentives in mobile crowd-sensing," *IEEE Access*, vol. 5, pp. 1382–1397, 2017.

[30] T. Luo, J. Huang, S. S. Kanhere, J. Zhang, and S. K. Das, "Improving iot data quality in mobile crowd sensing: A cross validation approach," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5651–5664, 2019.

[31] M. Monga and S. Sicari, "Assessing data quality by a cross-layer approach," in *2009 International Conference on Ultra Modern Telecommunications & Workshops*. IEEE, 2009, pp. 1–8.

[32] F. H. Bijarbooneh, W. Du, E. C.-H. Ngai, X. Fu, and J. Liu, "Cloud-assisted data fusion and sensor selection for internet of things," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 257–268, 2015.

[33] S. Madden *et al.*, "Intel lab data," *Web page, Intel*, 2004. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html

[34] A. C. Rai, P. Kumar, F. Pilla, A. N. Skouloudis, S. Di Sabatino, C. Ratti, A. Yasar, and D. Rickerby, "End-user perspective of low-cost sensors for outdoor air pollution monitoring," *Science of The Total Environment*, vol. 607, pp. 691–705, 2017.

[35] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[36] D. E. Boubiche, M. Imran, A. Maqsood, and M. Shoaib, "Mobile crowd sensing–taxonomy, applications, challenges, and solutions," *Computers in Human Behavior*, vol. 101, pp. 352–370, 2019.

[37] M. M. Rathore, A. Ahmad, A. Paul, and S. Rho, "Urban planning and building smart cities based on the internet of things using big data analytics," *Computer networks*, vol. 101, pp. 63–80, 2016.

[38] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE wireless communications*, vol. 24, no. 3, pp. 10–16, 2017.

[39] ISO 25000 Portal, "ISO/IEC 25012," 2019. [Online]. Available: https://iso25000.com/index.php/en/iso-25000-standards/iso-25012?start=0

[40] S. Sicari, A. Rizzardi, C. Cappiello, D. Miorandi, and A. Coen-Porisini, "Toward data governance in the internet of things," *Studies in Computational Intelligence*, vol. 715, pp. 59–74, 2018.

[41] S. Sicari, C. Cappiello, F. De Pellegrini, D. Miorandi, and A. Coen-Porisini, "A security-and quality-aware system architecture for Internet of Things," *Information Systems Frontiers*, vol. 18, no. 4, pp. 665–677, 2016.

[42] D. Kuemper, T. Iggena, R. Toenjes, and E. Pulvermueller, "Valid . IoT - a Framework for Sensor Data Quality Analysis and Interpolation," in *In MMSys'18: 9th ACM Multimedia Systems Conference*, 2018, p. 10.

[43] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for data quality assessment and improvement," *ACM Computing Surveys*, vol. 41, no. 3, 2009.

[44] J. Guo and F. Liu, "Automatic data quality control of observations in wireless sensor network," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 4, pp. 716–720, 2015.

[45] J. Liono, P. P. Jayaraman, A. K. Qin, T. Nguyen, and F. D. Salim, "QDaS: Quality driven data summarisation for effective storage management in Internet of Things," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 196–208, 2019. [Online]. Available: https://doi.org/10.1016/j.jpdc.2018.03.013

[46] A. Klein and W. Lehner, "Representing data quality in sensor data streaming environments," *Journal of Data and Information Quality*, vol. 1, no. 2, 2009.

[47] J. Byabazaire, G. O'Hare, and D. Delaney, "Data Quality and Trust : A Perception from Shared Data in IoT," pp. 1–6, 2020.

[48] R. Abo and A. Even, "Sampling density and frequency as data quality determinants in smart grids," *2017 Smart Cities Symposium Prague, SCSP 2017 - IEEE Proceedings*, 2017.

[49] C. C. Castello, J. Sanyal, J. Rossiter, Z. Hensley, and J. R. New, "Sensor data management, validation, correction, and provenance for building technologies," *ASHRAE Conference-Papers*, vol. 120, pp. 370–382, 2014.

[50] A. de Santander, "Santander Datos Abiertos." [Online]. Available: http://datos.santander.es

[51] A. Ruas, "Sense-city Demonstrator Dataset." [Online]. Available: http://dx.doi.org/10.25578/5M5SMI

[52] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/

[53] R. Tönjes, P. Barnaghi, M. Ali, A. Mileo, M. Hauswirth, F. Ganz, S. Ganea, B. Kjærgaard, D. Kuemper, S. Nechifor *et al.*, "Real time iot stream processing and large-scale data analytics for smart city applications," in *poster session, European Conference on Networks and Communications*. sn, 2014, p. 10.

[54] SIATA, "Sistema de Alerta Temprana de Medellín y el Valle de Aburrá," 2021. [Online]. Available: https://www.siata.gov.co/sitio_web/index.php/home

[55] U. E. P. A. EPA, *Quality Assurance Handbook for Air Pollution Measurement Systems*, 2017, vol. 2.

[56] PROANTIOQUIA, Universidad EAFIT, Fundación Corona, Comfama, Comfenalco Antioquia, Cámara de comercio de Medellín para Antioquia, El Colombiano, Cámara de comercio de Bogotá, and El Tiempo, "Medellín cómo vamos," 2020. [Online]. Available: https://www.medellincomovamos.org/node/18687

[57] E. UNION *et al.*, "Directive 2008/50/ec of the european parliament and of the council of 21 may 2008 on ambient air quality and cleaner air for europe," *Official Journal of the European Union*, 2008.

[58] JCGM, "Evaluation of measurement data — Guide to the expression of uncertainty in measurement," *International Organization for Standardization Geneva ISBN*, vol. 50, no. September, p. 134, 2008. [Online]. Available: http://www.bipm.org/en/publications/guides/gum.html

[59] E. W. Group, "Guide to the demonstration of equivalence of ambient air monitoring methods," 2010.

[60] A. Floris and L. Atzori, "Managing the quality of experience in the multimedia internet of things: A layered-based approach," *Sensors*, vol. 16, no. 12, p. 2057, 2016.

[61] D. Pal, V. Vanijja, and V. Varadarajan, "Quality provisioning in the internet of things era: Current state and future directions," in *Proceedings of the 10th International Conference on Advances in Information Technology*, 2018, pp. 1–7.

[62] D. Pal, V. Vanijja, C. Arpnikanondt, X. Zhang, and B. Papasratorn, "A quantitative approach for evaluating the quality of experience of smart-wearables from the quality of data and quality of information: An end user perspective," *IEEE Access*, vol. 7, pp. 64 266–64 278, 2019.

[63] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," *Communications of the ACM*, vol. 45, no. 4, pp. 211–218, 2002.

[64] M. del Pilar Angeles and F. GarcĂa-Ugalde, "Subjective assessment of data quality considering their interdependencies and relevance according to the type of information systems," *International Journal on Advances in Software*, vol. 5, no. 3, 2012.

[65] *Composite Index Number*. New York, NY: Springer New York, 2008, pp. 103–104. [Online]. Available: https://doi.org/10.1007/978-0-387-32833-1_73

[66] T. Saaty, "Relative measurement and its generalization in decision making why pairwise comparisons are central in mathematics for the measurement of intangible factors the analytic hierarchy/network process," *RACSAM-Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas*, vol. 102, no. 2, pp. 251–318, 2008.

[67] Zach, "Intraclass Correlation Coefficient: Definition + Example," 2021. [Online]. Available: https://www.statology.org/intraclass-correlation-coefficient/

Abbreviations

APH	Analytic Hierarchy Process
AQ	Air Quality Quality
CPS	Cyber-Physical Systems
DQ	Data Quality
DQI	Data Quality Indicators
EPA	Environmental Protection Agency
IoT	Internet of Things
DQO	Data Quality Objectives
PCM	Pairwise Comparison Matrix
PM	Particulate Matter
QoE	Quality of Experience
QoD	Quality of Data
QoI	Quality of Information
SIATA	Sistema de Alerta Temprana de Medellín y el Valle de Aburrá
TLS	Transport Layer Security
WSN	Wireless Sensor Network