Data quality issues in mobile crowdsensing environments

Souheir Mehanna

université PARIS-SACLAY

Data quality issues in mobile crowdsensing environments

Qualité des données dans les environnements de capteurs mobiles

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique
Graduate School : Informatique et sciences du numérique
Référent : Université de Versailles–Saint–Quentin–en–Yvelines

Thèse préparée dans l'unité de recherche **DAVID** (Université Paris–Saclay, UVSQ), sous la direction de **Mme Zoubida KEDAD**, Professeure et sous le co-encadrement de Mohamed CHACHOUA, Enseignant-Chercheur

Thèse soutenue à Versailles, le 11 Octobre 2023, par

Souheir MEHANNA

Composition du jury

Membres du jury avec voix délibérative

Amar RAMDANE-CHERIF Professeur, UVSQ - Paris Saclay	Prèsident
Isabelle COMYN-WATTIAU Professeure, CNAM / ESSEC	Rapporteur & Examinatrice
Abderrezak RACHEDI Professeur, Université Gustave Eiffel	Rapporteur & Examinateur
Malika GRIM-YAFSAH Enseignant-chercheur, Ecole Nationale des Sciences Géographiques (ENSG-géomatique)	Examinatrice

THESE DE DOCTORAT
NNT : 2023UPASG053

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Qualité des données dans les environnements de capteurs mobiles

Mots clés: Qualité des données, Données de capteurs, Complétude des données, Imputation des données manquantes, détection des anomalies

Résumé: Les environnements de capteurs mobiles sont devenus le paradigme de référence pour exploiter les capacités de collecte des appareils mobiles et recueillir des données variées en conditions réelles. Pour autant, garantir la qualité des données recueillies reste une tâche complexe car les capteurs, souvent à bas coûts et ne fonctionnant pas toujours de façon optimale, peuvent être sujets à des dysfonctionnements, des erreurs, voire des pannes. Comme la qualité des données a un impact direct et significatif sur les résultats des analyses ultérieures, il est crucial de l'évaluer.

Dans notre travail, nous nous intéressons à deux problématiques majeures liées à la qualité des données recueillies par les environnements de capteurs mobiles.

Nous nous intéressons en premier à la complétude des données et nous proposons un ensemble de facteurs de qualité adapté à ce contexte, ainsi que des métriques permettant de les évaluer. En effet, les facteurs et métriques existants ne capturent pas l'ensemble des caractéristiques associées à la collecte de données par des capteurs. Afin d'améliorer la complétude des données, nous nous sommes intéressés au problème de génération des données manquantes. Les techniques actuelles d'imputation de données génèrent les données manquantes en se reposant sur les données existantes, c'est à dire les mesures déjà réalisées par les capteurs, sans tenir compte de la qualité de ces données qui peut être très variable. Nous proposons donc une approche qui étend les techniques existantes pour permettre la prise en compte de la qualité des données pendant l'imputation.

La deuxième partie de nos travaux est consacrée à la détection d'anomalies dans les données de capteurs. Tout comme pour l'imputation de données, les techniques permettant de détecter des anomalies utilisent des métriques sur les données mais ignorent la qualité des ces dernières. Pour améliorer la détection, nous proposons une approche fondés sur des algorithmes de clustering qui intègrent la qualité des capteurs dans le processus de détection des anomalies.

Enfin, nous nous sommes intéressés à la façon dont la qualité des données pourrait être prise en compte lors de l'analyse de données issues de capteurs. Nous proposons deux contributions préliminaires: des opérateurs d'agrégation qui considère la qualité des mesures, et une approche pour évaluer la qualité d'un agrégat en fonction des données utilisées dans son calcul.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Data quality issues in mobile crowdsensing environments
Keywords: Data Quality, Sensor Data, Data Completeness, Data Imputation, Anomaly Detection

Abstract: Mobile crowdsensing has emerged as a powerful paradigm for harnessing the collective sensing capabilities of mobile devices to gather diverse data in real-world settings. However, ensuring the quality of the collected data in mobile crowdsensing environments (MCS) remains a challenge because low-cost nomadic sensors can be prone to malfunctions, faults, and points of failure. The quality of the collected data can significantly impact the results of the subsequent analyses. Therefore, monitoring the quality of sensor data is crucial for effective analytics.

In this thesis, we have addressed some of the issues related to data quality in mobile crowdsensing environments. First, we have explored issues related to data completeness. The mobile crowdsensing context has specific characteristics that are not all captured by the existing factors and metrics. We have proposed a set of quality factors of data completeness suitable for mobile crowdsensing environments. We have also proposed a set of metrics to evaluate each of these factors. In order to improve data completeness, we have tackled the problem of generating missing values. Existing data imputation techniques generate missing values by relying on existing measurements without considering the disparate quality levels of these measurements. We propose a quality-aware data imputation approach that extends existing data imputation techniques by taking into account the quality of the measurements.

In the second part of our work, we have focused on anomaly detection, which is another major problem that sensor data face. Existing anomaly detection approaches use available data measurements to detect anomalies, and are oblivious of the quality of the measurements. In order to improve the detection of anomalies, we propose an approach relying on clustering algorithms that detects pattern anomalies while integrating the quality of the sensor into the algorithm.

Finally, we have studied the way data quality could be taken into account for analysing sensor data. We have proposed some contributions which are the first step towards quality-aware sensor data analytics, which consist of quality- aware aggregation operators, and an approach that evaluates the quality of a given aggregate considering the data used in its computation.

Résumé substantiel en français

Les environnements de capteurs mobiles sont devenus le paradigme de référence qui exploite les capacités de collecte des appareils mobiles pour recueillir des données variées en conditions réelles à des fins d'analyse. Pour autant, garantir la qualité des données recueillies reste une tâche complexe car les capteurs, souvent à bas coûts et ne fonctionnant pas toujours de façon optimale, peuvent être sujets à des dysfonctionnements, des erreurs ou des pannes. Comme la qualité des données a un impact direct et significatif sur les résultats des analyses ultérieures, il est important de disposer d'outils d'évaluation et d'amélioration de la qualité. Dans notre travail, nous nous intéressons à deux problèmes liés à la qualité des données dans les environnements de capteurs mobiles, la complétude des données et la détection d'anomalies.

Notre première contribution porte sur la complétude des données. Nous nous sommes intéressés à l'évaluation de la complétude dans des environnements de capteurs mobiles, et étudié l'adéquation des facteurs de complétude existants dans ce contexte. Nous avons proposé trois facteurs de qualité : (i) la complétude temporelle, qui représente la façon dont une période de temps est couverte par les mesures de capteurs disponibles ; (ii) la complétude spatiale, qui représente la façon dont une zone géographique est couverte par les mesures de capteurs disponibles et (iii) la complétude d'un capteur qui représente la capacité du capteur à fournir les mesures attendues pour une période de référence. Nous avons proposé des métriques d'évaluation pour chacun de ces facteurs.

Afin d'améliorer la complétude des données, nous nous sommes intéressés au problème de génération des données manquantes. Les techniques existantes d'imputation de données génèrent les données manquantes en exploitant les données existantes, c'est à dire dans notre contexte les mesures déjà réalisées par les capteurs. Le résultat est donc très dépendant de la qualité des données utilisées pour l'imputation. Nous avons proposé des extensions d'approches d'imputation existantes qui intègrent la qualité des données lors de l'imputation, et nous avons en particulier étendu les approches ST-MVL, KNN-Impute et SVD-Impute.

Nous avons évalué les métriques de complétude ainsi que les approches d'imputation de données fondées sur la qualité dans le cadre du projet ANR Polluscope, sur des données réelles. L'objectif du projet est de quantifier l'impact de l'exposition individuelle à la pollution de l'air. Nous avons notamment montré que la prise en compte de la qualité lors de l'imputation permet d'améliorer de façon significative les résultats obtenus. Notre deuxième contribution porte sur la détection d'anomalies dans les données collectées dans des environnements de capteurs mobiles. Plusieurs catégories d'approches ont été proposées, comme les approches statistiques, les approches à base de clustering ou encore les approches fondées sur les réseaux de neurones. Toutes ces approches exploitent les données existantes pour identifier les données considérées comme des anomalies. Afin d'améliorer leur détection, nous avons proposé une approche fondée sur l'algorithme de clustering k-means. Notre approche décompose les séries temporelles collectées par les capteurs en séquences de taille fixe, puis regroupe ces séquences en clusters en tenant compte de la qualité des données ainsi que de certains éléments de contexte lorsqu'ils sont disponibles. Nous avons introduit les notions de qualité d'un cluster et de qualité d'une séquence, que nous utilisons pour assigner un score d'anomalie à chaque séquence, permettant de déterminer s'il s'agit d'une anomalie ou non. Nous avons réalisé des évaluations et montré que notre approche améliore la détection des anomalies comparée à des approches existantes fondées sur k-means.

Notre troisième contribution porte sur l'étude des approches possibles pour prendre en compte la qualité des données lors du calcul d'indicateurs à partir de données issues de capteurs. Nous avons proposé des opérateurs qui considèrent la qualité des données pendant le calcul d'agrégat. Nous avons défini deux façons de prendre en compte la qualité lors de l'agrégation : (i) la pondération des données, qui consiste à assigner à chaque donnée un poids représentant sa qualité, et (ii) le filtrage des données, qui consiste à considérer uniquement les données dont la qualité est supérieure à un seuil prédéfini lors de l'agrégation. Enfin, nous avons proposé une approche d'évaluation de la qualité d'un agrégat à partir de la qualité des données utilisées pour son calcul. Dans ce volet de notre travail, nous utilisé nos métriques de qualité précédemment définies, notamment pour la complétude des données et la qualité d'un capteur.

Contents

List of Figures

List of Tables

1 - Introduction

1.1 . Context and Motivation

The advent of the smart city concept has given rise to a profusion of innovative technologies using IoT (Internet of Things) [Zappatore et al., 2017], [Okafor et al., 2020] in several areas such as transportation, energy or environment. Indeed, the recent development of low-cost micro-sensor technology has inspired new fields of research and development in these areas [Joglekar and Kulkarni, 2016], [Zappatore et al., 2019], [Alvear et al., 2018], [Mehanna et al., 2020], [Dessimond et al., 2021]. The aim of this research is to respond to the problems raised by the growth of urban densification[1], requiring continuous adaptations of the urban system, particularly in terms of environmental quality, mobility and health safety. One of the key issues in all of these projects concerns the quality of data from sensors, whether fixed or mobile [Ehrlinger and Wöß, 2018], [Liu et al., 2019]. This is a fundamental issue because the reliability of the analyses performed on the data and the quality of the decision-making process depend essentially on data quality. This is precisely what our research is focusing on.

As mentioned above, the rise of nomadic sensor usage in various domains, such as medical monitoring equipment, banking transactions, road traffic surveillance, air pollution quantification, etc., increases the need for good-quality data. This growth generates vast volumes of data, necessitating their constant monitoring to ensure informed decision-making. Thus, in this context, data quality control is essential for the reliability of the results of the analyses and the associated decisions. However, these nomadic sensors are vulnerable which leads to various intrinsic malfunctions, such as calibration problems, battery, etc., and extrinsic ones, such as transmission or reception problems, misuse, etc. The loss of data and the presence of anomalies can happen during the acquisition or the integration process, due to several problems, such as the points of failure of the measuring sensors, the manufacturing defects in the sensors, and other losses of data chunks during the data integration process. Such issues can raise questions about the reliability of this data. Hence, ensuring a good quality data is an indispensable part of the data analysis process to enable informed decision-making. Thus, many research works are carried out in this direction [Zappatore et al., 2019], [Safaei et al., 2020], [Mehanna et al., 2020], [Thomas and J.E, 2021].

The goal of our work is to study data quality issues in mobile crowdsensing

[1]According to the World Bank, the urban population, currently 56%, will almost double by 2050: `https://www.banquemondiale.org/fr/home`.

environments. We are interested in the limitations of existing quality dimensions and factors in order to better capture the different facets of quality in this specific context. One of our goals is to propose new dimensions and factors and their associated metrics that are adequate for this mobile crowdsensing context. We also aim to propose approaches to improve the quality of sensor data in this context. Finally, we want to use these quality metadata in order to compute more reliable indicators and analyses. Similarly to the work of [Berti-Équille et al., 2011], we consider that data quality could be described by several dimensions. Each dimension has several factors that capture a particular quality facet. The quality factors are measured by instruments called metrics. For example, data accuracy is a quality dimension which could be refined into several quality factors. One of these factors is syntactic accuracy, which could be assessed using several metrics, one of them could be the comparison of the data to a regular expression defining the pattern this data must follow; this could be the general form of an email for example. In our work, we have focused on two data quality problems related to the missing values and the presence of anomalies. We propose some approaches to improve the quality of the data considering the data quality. We also initiate a work towards taking into account the quality while computing indicators from the data.

The work in this manuscript was done within the context of the ANR Polluscope project[2] [Brahem et al., 2021], [Languille et al., 2020], [Abboud et al., 2021]. The project lies in the context of environmental smart cities. The main objective of this project is to quantify human exposure to air pollutants. As human beings spend up to 60% of their daily time indoors [Languille et al., 2020], it does not seem reasonable to rely on stations that measure the air quality outdoors only for adequate human exposure. Reference machines are placed at a height of 2.5 meters, which is not at the same level as the height of an average human being. This means that reference stations do not target the actual human exposure. Hence, the main goal of the project is to have human carriers of mobile, low-cost sensor units that can be carried throughout the daily routines of individuals for a specific period to study their exposure to air pollutants. In addition to measuring air quality at the individual scale, it also offers the possibility of measuring air quality in indoor and outdoor environments. However, in the chain of data acquisition via micro-sensors, we are confronted with issues related to the completeness of the data, the presence of anomalies, and the introduction of data quality when computing aggregates on sensor data.

1.2 . Challenges

[2]For more information, please follow this link: `https://polluscope.uvsq.fr`

The advantages of recent low-cost nomadic sensor technology are numerous. In addition to cost, this technology is a real opportunity to facilitate data acquisition in many fields. However, for a variety of reasons, data (measurements) from this type of sensor are likely to have imperfections (inaccuracy, inconsistency, etc.). In other words, the quality of such data is likely to be impaired. These data quality issues can result from several possible problems that happen during either the acquisition or the integration processes. Missing values and anomalies are the most frequent problems in mobile crowdsensing environments. Missing values have been studied in the related works on data quality and are related to the data completeness dimension. The presence of anomalies in the data is related to another dimension of data quality which is the accuracy. We have focused in our work on the problem of missing values and the presence of anomalies in the data.

Many definitions and metrics exist in the literature on data completeness. However, mobile crowdsensing environments have specific characteristics that are not all captured by these metrics. Hence, the existing metrics are not suitable to capture all the characteristics of data completeness in this context. For example, an existing metric of completeness is the proportion of null values in the dataset. However, there are other factors of completeness in mobile crowdsensing environments. For example, we could be interested in the extent to which our measurements cover a specific geographical area, and the proportion of null values would fail to capture this facet of the data.

In order to improve the completeness of the data, one solution is to generate the values for the missing ones in the dataset. There are numerous approaches targeting the generation of missing values using data imputation techniques for sensor data. However, the replacement of missing values is still a challenging task. Data imputation techniques rely on existing data to generate a missing measurement value. The problem with existing approaches is that they neglect that the sensors may not always operate in an optimal way, which may result in poor quality data. This means that measurements coming from sensors that performed poorly due to some reason are considered in the same way as those coming from a sensor that is performing optimally. This results in imputed values that are of poor quality.

Another major problem faced when dealing with sensor data is the presence of undetected anomalies. Anomalies are data points or sequences that do not conform to the normal behavior observed in the data. They could manifest as spikes, unusual points, or unusual patterns that often reveal interesting information in the data. To this day, detecting pattern anomalies remains challenging. There is a wide variety of approaches targeting anomaly detection [Khayati et al., 2020], [Braei and Wagner, 2020], [Blázquez-García et al., 2021]. Many of these approaches rely on existing data measurements to identify anomalies. This means that the quality of the detection results also relies on the quality of this input

data. Given the vulnerable nature of the emerging low-cost nomadic sensors to errors, anomaly detection techniques could deal with measurements of various qualities. Therefore, a major challenge is improving the quality of these anomaly detection approaches by considering the quality of the underlying measurements.

Given a dataset and a set of tools to measure its quality, another challenge is to take data quality into account in the computation of indicators. As the quality of the measurements used to compute an aggregate is not always optimal, another challenge is assessing the quality of this aggregate given the quality values of the data measurements used in the computation.

1.3 . Contributions

In our work, we have proposed some contributions related to the data completeness, the presence of anomalies in the data, and computing indicators considering data quality. Our contributions are the following.

- We define a set of data completeness factors dedicated to mobile crowdsensing environments. We identify three data completeness factors suitable for this context: sensor completeness, spatial completeness, and temporal completeness. We characterize and propose the associated metrics to assess these factors.
- We propose a quality-aware data imputation technique in order to take data quality into account during the generation of the missing values. We extend three data imputation techniques: ST-MVL, SVDImpute, and KNNImpute. We consider that the data quality is captured by aggregating different quality facets related to the device, user behavior, and reference datasets. We evaluate the proposed approach on real data from the Polluscope project [3] [Brahem et al., 2021], [Languille et al., 2020] and prove that the results of the data imputation are improved when using data quality for the imputation.
- We propose a quality-aware anomaly detection approach that detects anomalies while taking into account the quality of the data measurements. The data quality and other relevant contextual information help understand the surrounding environment, making it easier to detect anomalies tied to their surrounding context. Our approach is based on clustering methods and targets pattern anomalies, which are more challenging to detect according to [Braei and Wagner, 2020].
- We propose a first contribution towards quality-aware sensor data analytics, consisting of some basic building blocks. We propose quality-aware aggregation operators that take quality into account to aggregate data measure-

[3] http://polluscope.uvsq.fr

ments, and we also propose an assessment method to compute the quality of the aggregates.

1.4 . Outline of the thesis

The remaining of this manuscript is organized into five chapters.

In chapter 2, we present some related works on existing data quality dimensions for traditional databases and the data quality dimensions proposed for sensor data. We survey works defining and proposing metrics for the evaluation of different data quality dimensions and specifically for data completeness. We identify several open research problems: the generation of missing values in mobile crowdsensing environments, anomaly detection, and specifically the challenge of identifying pattern anomalies that could reveal interesting knowledge about the measured element once detected. Also, improving existing data aggregation operators and assessing the quality of aggregates.

Chapter 3 is devoted to the assessment and improvement of data completeness in mobile crowdsensing environments. We characterize three factors for data completeness that are relevant in mobile crowdsensing environments (MCS). We propose metrics to assess these different factors. We present our approach to improving the data completeness quality dimension by proposing a quality extension to three existing data imputation techniques to generate the missing values in the dataset using the quality of the data measurements.

In chapter 4, we present our quality-aware anomaly detection approach. The approach injects data quality into the process of identifying anomalies. The quality of the data and some contextual information are employed to group similar subsequences together and to compute an anomaly score taking into account the quality of the measurements.

Chapter 5 is a first step towards a quality-enhanced data integration system. The idea is that when computing indicators and when performing analyses on the data, we need to take into account the quality of the sensor that captured the measurements because analysis based on poor quality data will lead to poor quality indicators. In this chapter, we present the way data quality could be used to improve sensor data analytics by introducing quality-aware aggregation operators that consider data quality in the aggregation process. We also present a method to compute the quality of a given aggregate.

Finally, in chapter 6, we conclude the manuscript by first summarizing our contributions, and then presenting some perspectives and future research directions.

2 - State of the Art

2.1 . Introduction

With the development of micro-sensor technologies and the diversification of data sources, the data quality problem has grown and has given rise to numerous research projects in different application areas [Batini and Scannapieco, 2006, Sidi et al., 2012, Wang et al., 2016]. Data quality (DQ) is defined by [Hassany Shariat Panahy et al., 2013] as fitness for use and as conformance to requirements [De Feo, 2017]. Several approaches address data quality issues for general contexts as well as for mobile crowdsensing contexts. There is no agreement on a standard definition of data quality that can be applied across all data domains [Jesiļevska, 2017]. In the context of mobile crowdsensing environments, there are numerous works that target different quality issues. However, there is no consensus in the definition of quality factors, which may be overlapping and contradicting [Peralta, 2006].

In this chapter, we study several research issues related to data quality in mobile crowdsensing environments. Many existing works defined the quality factors of sensor data [Östman, 1997], [Rodríguez and Servigne, 2013]. Some works focused on the definition and characterization of data quality in this context and the identification of the different quality dimensions. Other works focus on the evaluation and assessment of the quality factors and their dimensions or the quality of service as provided by the sensors such as [Serhani et al., 2016], [Biswas et al., 2006]. Other works focus on improving the quality of the data from several aspects and dimensions. We present in this chapter the existing data quality factors and discuss their applicability to mobile crowdsensing environments. We also discuss approaches that improve the quality of the data in this context. Our work is focused on studying two data quality problems, the missing values and the presence of anomalies. Missing values are related to data completeness quality dimension and the anomalies are related to data accuracy. Several works define the data completeness quality factor in different contexts [Klein et al., 2007], [Liu et al., 2017], [Azimi and Pahl, 2021], and others propose metrics to evaluate it [Biswas et al., 2006], [Todoran et al., 2015], [Ehrlinger and Wöß, 2022]. We also study the improvement of data completeness by exploring approaches that generate missing values using data imputation techniques such as [Troyanskaya et al., 2001],[Yi et al., 2016],[Khayati et al., 2020].

Anomaly detection has been the focus of several works. Some approaches have worked on defining anomalies in time series and proposed techniques to detect the different types of anomalies [Gupta et al., 2014], [Braei and Wagner, 2020],

[Blázquez-García et al., 2021]. Some approaches also discussed contextual anomaly detection in the presence of knowledge about the surrounding circumstances and conditions of the sensors collecting the measurements [Tsay et al., 2000], [Liang and Parthasarathy, 2016],[Zheng et al., 2017].

Finally, we have also reviewed some existing works that include data quality information in the data integration process in order to analyze the impact of data quality on the analysis process and on the computation of indicators from the data. We have studied approaches that incorporate data quality for spatiotemporal data [Boulil et al., 2013], [Berrahou et al., 2015], for data collected within a mobile crowdsensing context [Huang et al., 2022] and for spatial data in SOLAP systems [Devillers et al., 2007a] because we are interested in ways to integrate quality of the data in the integration process within a mobile crowdsensing context.

The remainder of this chapter is organized as follows. Quality dimensions and data quality assessments are presented in section 2.2. Section 2.3 discusses the quality dimensions specific to mobile crowdsensing environments (MCS). In section 2.4, we discuss data imputation approaches. Section 2.5 presents approaches that studied the detection of anomalies. Section 2.6 discusses quality-aware data integration approaches. Finally, section 2.7 concludes the chapter.

2.2 . Data quality definition and assessment

Many works have defined data quality dimensions for different application domains and contexts of data. The work of [Berti-Équille et al., 2011] proposes quality abstraction composed of data quality dimensions, quality factors, and quality metrics. Each dimension has several factors, and a quality metric measures a quality factor. Many existing approaches propose systems that monitor the quality of the data and metrics to evaluate the different quality dimensions. The authors of [Alizamini et al., 2010] define the data quality dimensions and propose the use of fuzzy association rules to measure data accuracy. Another definition for data quality is fitness for use [Hassany Shariat Panahy et al., 2013], [Sidi et al., 2012]. Each dimension captures a specific aspect of the quality of the data [Batini and Scannapieco, 2006]. The authors of [Sidi et al., 2012] defined a data quality dimension as a characteristic or part of the information that provides a way for measuring and managing data quality. The work of [McGilvray, 2021] states that a data quality dimension offers a way for measuring and managing data quality as well as information.

In this section, we present some existing dimensions defined in the data quality management community that are not necessarily specific to mobile crowdsensing environments. We then present metrics that have been proposed for quality as-

sessment of several data quality dimensions. We finally provide an analysis of these definitions and metrics proposed.

2.2.1 . Data quality dimensions

Many works have defined data quality dimensions for different contexts. This section presents existing definitions of some data quality dimensions. Some approaches worked on grouping the data quality dimensions into categories. The authors of [Wang and Strong, 1996] defined and categorized the dimensions of data quality according to four categories, (1) *intrinsic*, (2) *accessibility*, (3) *contextual*, and (4) *representational data quality*, as described below:

- Intrinsic: captures the quality that data has on its own. The following four quality dimensions: *accuracy*, *objectivity*, *believability*, and *reputation* are classified under this category because they capture the intrinsic aspect of the data.
- Accessibility: refers to the extent to which data are available or obtainable. The following data quality dimensions are classified under this category: *accessibility* and *access security*.
- Contextual: highlights the requirement that data quality must be considered within the context of the task at hand;. The following data quality dimensions are classified under this category: *relevancy*, *value-added*, *timeliness*, *completeness*, and *amount of data*.
- Representational: describes how understandable and representative the data is. The following data quality dimensions are classified under this category: *interpretability*, *ease of understanding*, *conciseness of representation*, and *consistency of representation*.

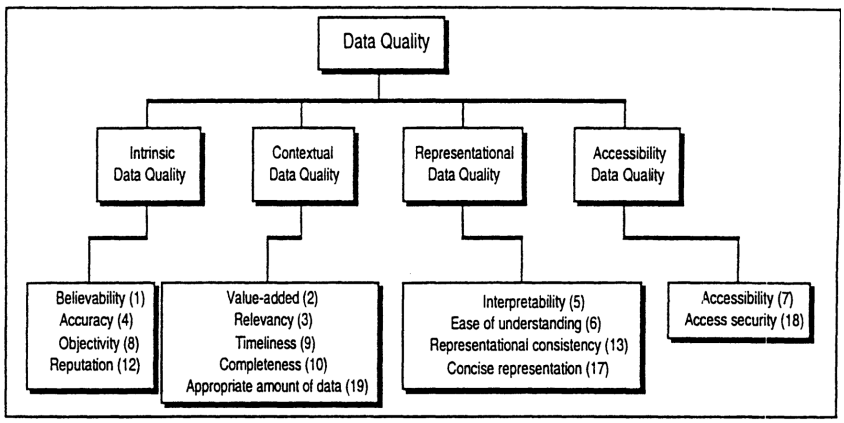

Figure 2.1: A typology of some data quality dimensions proposed by [Wang and Strong, 1996]

The authors of [Wang and Strong, 1996] proposed a possible typology of data quality dimensions shown in Figure 2.1. Several other works have defined different typologies. In what follows, we present some data quality dimensions in the literature from each one of these categories.

Accuracy

Several works have defined data accuracy in the literature. The authors of [Wang and Strong, 1996] defined accuracy as the extent to which data is correct, reliable, and certified. Some works have defined data accuracy compared to some ground truth or reference. It has been defined by [Batini and Scannapieco, 2006] as the closeness between a value v and a value $v\prime$, considered the correct representation of the real-life phenomenon that v aims to represent. Data accuracy measures the degree of similarity between the actual data and the ground truth [Alfred., 1972]. The ISO 25000 standards for data quality [iso25000,] define accuracy as the degree to which data has attributes that correctly represent the true value of the intended feature or event in a specific context of use. The authors of [Ehrlinger and Wöß, 2022] defined accuracy as the magnitude of an error. The magnitude of error in this definition means the value of the difference between the actual data and the ground truth. The authors of [McGilvray, 2021] also considered accuracy as a measure of the correction of the data, which requires an authoritative source of reference to be identified and accessible. [Östman, 1997] defined accuracy as the closeness of observation to true values or values accepted to be true. Data accuracy refers to the degree to which the primary data differ from the population's true parameters determined by using established secondary data sources [Nonnemacher et al., 2014, Jacke et al., 2012]

Some works have defined different factors of data accuracy. The authors of [Batini and Scannapieco, 2006] identified two factors of accuracy: $syntactic$ and $semantic$. $Syntactic\ accuracy$ is defined as the closeness of a value v to the elements of the corresponding definition domain D. It is measured by the number of syntactic operations that need to be applied to a value v to convert it to another element belonging to the domain of definition. For example, the syntactic accuracy of a registered customer name in a database "Jhon" takes one syntactic operation by moving one letter to be converted to the correct value which is "John". The authors of this work also define $semantic\ accuracy$ as the closeness of a value v to the true value $v\prime$.

The work of [Östman, 1997] identified three factors of data accuracy: $positional$, $temporal$, and $thematic$. The authors defined positional accuracy as a quality parameter indicating the accuracy of geographic positions. The authors later distinguished two types of accuracy: $structural$ and $temporal$. The authors referred to $temporal\ accuracy$ as the rapidity with which the change

in the real-world phenomenon is reflected in the update of the data value [Batini and Scannapieco, 2016]. This type of accuracy measures how fast the data is updated in the information system once a change happens in the real world. The other type is *structural accuracy* which characterizes the accuracy of data as observed in a specific time frame, where the data value can be considered as stable and unchanged [Batini and Scannapieco, 2016].

Data Completeness

The problem of data completeness has been known in the literature as the problem of missing information [Emran, 2015]. Data completeness is related to the presence and the absence of the data in an information system.

Several works have defined the meaning of data completeness. The authors of [Wang and Strong, 1996] defined completeness as the extent to which data are of sufficient breadth, depth, and scope for the task at hand. ISO 25000 [iso25000,] defined data completeness 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. The authors of [Batini and Scannapieco, 2006] defined data completeness in relational databases as the extent to which the table represents the corresponding real world. For example, considering a transactions data table containing different information about bank transactions, data completeness is defined as how representative the data in this table is of the actual transactions that happened at the bank. Data completeness within an entity has been defined as data for which all required information is present [Bovee et al., 2003]. According to [Fox et al., 1994], completeness was defined as the degree to which a data collection has values for all attributes of all entities. The latter definition was proposed in the context of relational data, where the completeness of a table is related to the completeness of all the records and the attributes in this table. The authors of [Nonnemacher et al., 2014] have defined data completeness for health data in databases as the degree to which the data have captured all relevant patients in accordance with the inclusion criteria. In the healthcare context, the completeness of the data covers all the expected data about relevant patients. The work of [Östman, 1997] defined completeness for spatial data as the degree of conformance of a geographic dataset compared to its nominal ground with respect to the presence of objects, association instances, and property instances.

Another set of works has defined data completeness on different levels for relational database contexts. The authors of [Liu et al., 2016] have defined data completeness over several parts: *attribute*, *tuple*, and *relation*. *Attribute completeness* is defined as the extent to which an attribute cell stores information. *tuple completeness* is defined as the extent to which all the cells of the tuple

store information. Finally, the authors defined *relation completeness* as the extent to which a relation describes entities of interest [Liu et al., 2016]. The work of [Pipino et al., 2002] identified two data completeness factors *column completeness*, and *population completeness*, defined as follows.

- **Column Completeness:** a measure of the missing values for a specific property or column in a table.
- **Population Completeness:** evaluates missing values with respect to a reference population.

Data freshness

The work of [Peralta, 2006] relates data freshness to how old data is and states that it is a quality dimension that represents a family of quality factors, each representing some freshness aspect and having its own metrics. The author of this work then distinguishes two quality factors for this quality dimension: *timeliness* and *currency*.

Some works, such as [Batini and Scannapieco, 2006], have defined timeliness as a factor that shows how current the data are for the task at hand. The authors of [Wang and Strong, 1996] defined timeliness as the extent to which the age of the data is appropriated for the task at hand. The author of [Peralta, 2006] characterizes timeliness as a quality factor that describes how old is the data. It captures the gap between the data creation/update and data delivery no matter when the data was extracted from the sources. It is often estimated as the time elapsed from the last update to a source [Peralta, 2006]. ISO standard [iso25000,] defined timeliness as the degree to which data has attributes that are of the right age in a specific context of use.

Age or *currency* is defined by [Bovee et al., 2003] as a measure of how old the information is, based on how long ago it was recorded. A datum is said to be current or up-to-date at time t according to [Fox et al., 1994] if it is correct at time t, and considered not up-to-date if it is incorrect at time t but was correct at some moment preceding t. For example, assume there is a table describing the salary of employees, if an employee gets a raise but their salary in the table is not updated, then the data is not up-to-date.

[Batini and Scannapieco, 2006] defined currency as how frequently data is updated. Currency is considered high when the data is up-to-date [Batini and Scannapieco, 2006]. It captures how promptly data are updated with respect to changes occurring in the real world [Batini and Scannapieco, 2016]. Currency describes how old data is with respect to the sources according to [Segev and Fang, 1989]. It captures the gap between the extraction of data from the sources and its delivery to the users. It is often measured as the time elapsed

since data was extracted from the source [Peralta, 2006].

The authors of [Langefors and Sundgren, 1975] suggest that it is required for each datum to have a time indicator. The authors of [Fox et al., 1994] recognize the special relationship between change over time and data quality and propose the definition of the following: *currency*, *age*, and *timeliness* that take into consideration the change of data over time.

Volatility is another data quality dimension that is relevant to *timeliness* and *currency*. The authors of [Bovee et al., 2003] described *volatility* of information as a measure of information instability. It is the frequency of change of the value for an entity attribute. Non-volatile information are stable, they do not change nor become dated. *Volatility* characterizes the frequency with which data vary in time [Batini and Scannapieco, 2006], [Batini and Scannapieco, 2016].

Consistency

Data consistency is a data quality dimension that refers to whether the data match certain rules or constraints. The authors of [Wang and Strong, 1996] define consistency as the extent to which data is presented in the same format and consistently represented and formatted. [iso25000,] defined consistency 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.

According to [Batini and Scannapieco, 2006], consistency captures the violation of semantic rules defined over (a set of) data items, where items can be tuples of relational tables or records in a file.

The authors of [Wang and Strong, 1996] have categorized data consistency under the representational data quality category. Representational data quality captures aspects related to the quality of data representation, such as interpretability. The work of [Bovee et al., 2003] proposes a model of information quality, its attributes, and their respective sub-attributes. This work has listed the data consistency quality dimension under the *integrity* category that implies that the data is free from defects or flaws or has the state of being unimpaired or sound [Bovee et al., 2003].

Synchronization is a quality dimension that is relevant to consistency and expresses how data is aligned and consistent across different systems. Some works in the literature associate it with data consistency such as [McGilvray, 2021]. The authors of [McGilvray, 2021] define consistency and synchronization in the context where the data is replicated among several data stores, applications, or systems. It is defined as the extent to which the data stored across these various systems is equivalent.

Interpretability, Accessibility

Many other data quality dimensions have been proposed and defined in the literature. We previously discussed the quality dimensions of accuracy, completeness, timeliness, and consistency identified by [Donoghue et al., 2011] as the four most important data quality dimensions. However, other quality dimensions exist, such as *interpretability* and *accessibility*.

The authors of [Batini and Scannapieco, 2006] defined interpretability as a dimension that is related to the documentation and metadata available that help us understand and interpret the meaning and properties of the data. The authors of [Bovee et al., 2003] defined the interpretability of data as data that is understandable, and data that we are able to derive meaning from. [Wang and Strong, 1996] defined understandability as the extent to which data are clear without ambiguity and easily comprehended. The work of [Batini and Scannapieco, 2016] defined interpretability as the ability of the user to correctly interpret values from their format.

Data accessibility is a quality dimension that refers to the degree to which the data is available to authorized users. As defined by [Batini and Scannapieco, 2006], accessibility is the ability of the user to access data from his culture, physical status, and available technologies. [Wang and Strong, 1996] defined accessibility as the extent to which information is available or easily and quickly retrievable. The authors of [Batini and Scannapieco, 2016] defined accessibility in data as data that is available or easily and quickly retrieved.

2.2.2 . Data quality assessment

Many research works have dealt with data quality assessment. This subsection presents some of the metrics from the literature that propose the assessment and evaluation of several data quality dimensions such as: accuracy, completeness, consistency, freshness, etc. In this subsection, we will focus on quality assessment in general, not only the ones that are specific to mobile crowdsensing environment.

Accuracy

Some works proposed metrics to assess data accuracy. The authors of [Jacke et al., 2012] proposed metrics to evaluate the accuracy of health data collected within a breast cancer study. The authors propose to evaluate the accuracy of this data by comparing all available risk, prognostic, and predictive factors to distributions available from external reference databases. The authors of [Östman, 1997] surveyed existing works for data quality assessment metrics and found that the most used metric to evaluate temporal accuracy was the date of

the last update. In the work of [Redman, 2005], the authors proposed to evaluate field-level and record-level accuracy as follows:

$$\text{field level accuracy} = \frac{\text{number of correct fields}}{\text{number of fields tested}}$$

The field-level accuracy is defined at the attribute level. The authors also proposed to assess record-level accuracy with respect to the number of records found to be accurate as follows:

$$\text{record level accuracy} = \frac{\text{number of accurate records}}{\text{number of records tested}}$$

Completeness

Numerous works have proposed metrics to evaluate the data completeness quality dimension. The work of [Östman, 1997] surveyed some quality dimensions and the metrics to evaluate them on spatial data such as techniques for the assessment of the data completeness quality dimension; the most used evaluation metric of data completeness was the percentage of missing objects in the data.

The work of [Köpcke et al., 2013] aimed to evaluate the completeness of electronic health record (EHR) data for the purpose of patient recruitment into clinical trials. The study focused on evaluating the presence of 16 data elements that are commonly required for patient recruitment into clinical trials, including demographics, diagnoses, procedures, medications, and laboratory test results. The authors matched the desirable patient characteristics to specific elements in the EHR data. This means that one patient characteristic is composed of several elements found in the EHR data. They defined corresponding data elements of a patient characteristic as those fields in the EHR's database which hold for at least one patient the information whether the patient has the characteristic or not. Data completeness was evaluated by the authors of [Köpcke et al., 2013] as the fraction of patient characteristics with at least one corresponding data element, multiplied by the fraction of patients with any data in at least one corresponding element.

The authors of [Xiao et al., 2017] developed a data quality evaluation tool that evaluates data completeness for health data collected from records at hospitals. The authors identified 34 variables in the recorded data and evaluated data completeness as the ratio of the number of available variables that had a value recorded out of the total 34 variables.
The work of [Batini and Scannapieco, 2006] considered that data completeness is

evaluated as the ratio of presence/absence of null values while considering different granularity levels: the value, tuple, attribute, and relation levels.

A generic metric for data completeness was proposed by the authors of [Ehrlinger and Wöß, 2022] based on the literature. The authors refer to an element as any data unit, such as an attribute, a record, or a table. The completeness metric is as follows:

$$\text{Completeness} = \frac{|e_C|}{|e|}$$

Where e_C is the number of complete elements and |e| is the total number of elements.

The work of [Lee et al., 2009] stated that completeness could be viewed from at least three perspectives: schema completeness, column completeness, and population completeness. The authors of this work proposed that each of these completeness perspectives can be measured by the following simple ratio:

$$\text{Completeness} = 1 - \frac{\text{Number of incomplete items}}{\text{Total number of items}}$$

Timeliness, Currency, Volatility

A set of works have assessed time-dependent data quality dimension such as volatility and factors such as timeliness, and currency. A timeliness definition has been proposed by the authors of [Bernd et al., 2007] that was later used and extended by the work of [Heinrich and Klier, 2009] to propose the following.

$$Q^{w}_{Time}(t) = exp(-decline(A).t)$$

Where w is the attribute value and $decline(A)$ is the decline rate specifying the average number of attributes that become outdated within the time period t.

The authors of [Ballou et al., 1998] postulate that the timeliness of the data depends upon when it is delivered to the consumer. It is evaluated as a function of the ratio of currency and volatility. The currency is the overall age of a data unit and is good or bad depending on the volatility, also called the shelf-life, of the data unit. Volatility is defined as the length of time data remains valid. A large value for currency is unimportant if the volatility/shelf-life is infinite. On the other hand, a small value for currency can deteriorate timeliness if volatility is very short. Therefore, is evaluated as follows:

$$\text{Timeliness } = max\{0, 1 - \frac{\text{currency}}{\text{volatility}}\}$$

In some cases, data can be very up-to-date and have a high currency, while at the same time, it might exhibit low volatility if the data values remain relatively stable

or do not change significantly over time. In such cases, timeliness equals zero. The authors of [Ballou et al., 1998] also proposed to evaluate data currency as follows:

$$\text{Currency} = \text{Age} + (\text{Delivery Time} - \text{Input Time})$$

Where Age measures how old the data is when received, Delivery Time is the time when the data was delivered, and Input Time is the time the data was obtained.

Consistency

The authors of [Lee et al., 2009] defined several perspectives of data consistency. One of these perspectives is the consistency between two related data elements, such as the name of a city and the postal code that must be consistent. Another perspective of data consistency identified by [Lee et al., 2009] is the consistency of format for the same data element used in different data tables. The authors proposed to evaluate consistency as follows:

$$\text{Consistency} = \frac{\text{Number of instances violating specific consistency type}}{\text{Total number of consistency checks performed}}$$

The work of [Fox et al., 1994] found that a possible measure of the consistency of an individual datum is a binary indication (Yes or No) of whether the datum satisfies all constraints. This can be extended into a measure for an entire data collection by determining the fraction of inconsistent data.

The authors of [Hinrichs, 2002, Ehrlinger and Wöß, 2022] proposed to evaluate the consistency of an attribute *w* as follows:

$$Q(w) = \frac{1}{\sum_{j=1}^{n} r_j(w).g_j + 1}$$

Where $r_j(w)$ is the violation of consistency rule r_j applied to attribute w, and g_j is the degree of severity of $r_j(w)$. The value of $r_j(w)$ is assigned according to the following:

$$r_j(w) \begin{cases} 0 & \text{if } w \text{ satisfies } r_j \\ 1 & \text{otherwise} \end{cases}$$

Other data quality dimensions

The authors of [Lee et al., 2009] proposed several other metrics to evaluate different quality dimensions. The accessibility of data reflects the ease of attainability of the data. The authors proposed a metric that emphasizes the time aspect of accessibility as follows:

$$\text{Accessibility} = max[(1 - \frac{I_{\text{Request-Delivery}}}{I_{\text{Request-No longer of use}}}), 0]$$

Where $I_{\text{Request-Delivery}}$ is the interval from request time to delivery time to the user, and $I_{\text{Request-No longer of use}}$ is the interval from request time to time at which the data is no longer of use. In some cases, data is no longer of use or outdated, while data is being requested and delivered to users. In such cases, the accessibility of the data is close to zero.

The authors of [Lee et al., 2009] also evaluated the *appropriate amount of data* quality dimension. This dimension reflects a state where the amount of data is neither too little nor too much. Their proposed metric for this dimension is as follows:

$$\text{Appropriate amount of data} = min[\frac{\text{Nb of data units provided}}{\text{Nb of data units needed}}, \frac{\text{Nb of data units needed}}{\text{Nb of data units provided}}]$$

2.2.3 . Analysis

In this section, we discussed definitions of data quality dimensions and their corresponding evaluation metrics proposed within contexts that are not necessarily mobile crowdsensing. We studied several works defining some well-known data quality dimensions.

Data accuracy has been frequently defined as the closeness of the data to the ground truth. Among the works we studied, two metrics were proposed to measure data accuracy. The first one is by comparing the value of the data to another in a reference dataset. The second one is by computing the fraction of data fields or records that are correct.

Data completeness has always been associated with missing values in the data. It is evaluated with the presence or absence of a value for a given characteristic in the dataset. Some works computed it using the percentage of missing values and others with the ratio of available data. Even though some of the works were defined within healthcare contexts, the metrics proposed are still applicable to other application domains.

Timeliness is a time-dependent data quality dimension associated with how recent and up-to-date the data are for tasks at hand. It has been evaluated using the rate at which data attributes become outdated with time.

Consistency was associated with (i) certain consistency rules/constraints on the data such as the format, and (ii) the consistency of the same data at different places, machines, or tables. It has been evaluated with the number of instances of data violating or conforming to the defined consistency rules.

The timeliness evaluation definition and proposed metrics study how the data age. The time dimension is an intrinsic characteristic in mobile crowdsensing data. Once a data measurement is registered, its value does not change, only new

measurements with new values are captured and recorded.

The proposed metrics for data accuracy, completeness, and consistency are directly applicable to mobile crowdsensing environments. However, they sometimes fail to capture the quality of data in the MCS context. If we consider the data completeness quality dimension of a set of data measurements as an example, even if the set has no null values and no attributes in the existing measurements are missing, the set could still be missing full records. Hence, these existing metrics are applicable and relevant to mobile crowdsensing environments but they are not sufficient.

Anomalies, being data points that deviate significantly from the norm, have the potential to introduce errors and inaccuracies into datasets. Detecting anomalies enables the identification of erroneous or anomalous data points and patterns that may arise from measurement errors, sensor malfunctions, or other irregularities. Addressing these anomalies can improve data accuracy. Improved data accuracy translates to more reliable analysis, robust decision-making, and the ability to extract meaningful insights from the data.

2.3 . Data quality in mobile crowdsensing environments

In the previous section, we have described existing definitions and metrics of data quality dimensions in contexts that are not necessarily mobile crowdsensing. The rise of sensing technologies raises the question of the applicability of these traditional dimensions to the MCS context. This section focuses on definitions and metrics of data quality dimensions applied to mobile crowdsensing environments. We study existing works that define data quality dimensions in sensing contexts. We then examine approaches that propose metrics to evaluate different quality dimensions in this context. Finally, we present our analysis of these proposed definitions and metrics.

2.3.1 . Data quality dimensions in MCS

Many works have defined different quality dimensions in mobile crowdsensing environments and sometimes to specific application domains [Juddoo et al., 2018, Serhani et al., 2016, Klein et al., 2007, Liu et al., 2019, Emran, 2015]. The work of [Juddoo et al., 2018] investigated the applicability of existing data quality dimensions to healthcare data in a big data context and surveyed the existing works studying data quality dimensions for healthcare. The work of [Serhani et al., 2016] discussed enforcement of data quality of healthcare data in a big data context. They also identified metrics for their evaluation. The authors of [Klein et al., 2007] defined sensor data quality in a smart environment. The authors of [Liu et al., 2019] discuss data quality problems and dimensions for sensor data in the context of the Internet of Things. The author of [Emran, 2015] conducts a review of the literature on the definitions of data completeness and the associated metrics. We discuss in this section the definitions of data qual-

ity dimensions applicable in mobile crowdsensing environments (MCS) and their corresponding evaluation metrics.

Accuracy

Data accuracy is defined in mobile crowdsensing contexts as a dimension that describes the numerical precision of a data value and states the absolute or relative error of a physical value [Klein et al., 2007]. It could be affected by several issues such as the lack of the sensor taking the measurement or the loss of calibration of this sensor or even low accuracy especially when the sensors are of low cost. There are numerous works that studied data accuracy and its importance in mobile crowdsensing environments. The work of [Juddoo et al., 2018] found that data accuracy was the quality dimension that was most cited among the works they studied.

The authors of [Rodríguez and Servigne, 2013] defined accuracy for sensor data in an environmental context as the correctness of data according to a reference value and sensor technical precision. [Liu et al., 2019] stated that data accuracy is the extent to which observation of the object truly reflects its real-world situation. For example, in the context of health data monitoring the heart rate, the accuracy of this data represents how close the obtained measurements are to the actual heart rate being measured.
The work of [Serhani et al., 2016] stipulates that data accuracy measures how much the recorded data is correct and hence is reliable.

Completeness

Data completeness in mobile crowdsensing environments is a dimension that expresses the extent to which all expected data is provided by IoT services [Liu et al., 2019]. Data completeness in this context could b[]e affected by the occasional data losses by the sensors, the issues in the integration process such as losses of measurements during the merging process of data from different sources, or any technical issues with the acquisition mechanism of the sensor.

Many approaches defined data completeness for different application domains within a mobile crowdsensing context. The authors of [Serhani et al., 2016] mainly defined data completeness to be related to the existence of missing or null values. According to [Klein et al., 2007], data completeness addresses the problem of missing values due to sensor failures or malfunctions. Some sensor failures or malfunctions could be the battery drain which causes the sensor to shut down, leading to some measurement losses while the sensor is down. The work of [Todoran et al., 2015] defines completeness as a dimension that measures the presence of all values for all the variables. The authors of [Fizza et al., 2022] defined sensor data completeness as the degree to which sensor data values are not missing for a given time window. The authors of [Fishbain et al., 2017] defined in their

toolkit a quality dimension called the presence, which is a quality dimension that measures the sensor's or the system's availability of a measurement at a given time.

Consistency

Consistency can be affected by factors such as sensor calibration, data processing methods, and user behavior. It is a data quality dimension that reflects the extent to which data are of the same format and respect some consistency rules defined on the data. Logical consistency was defined by the authors of [Östman, 1997] as the degree of conformance of a geographical dataset with respect to the constraints defined in the application schema. [Liu et al., 2019] defined concordance, which is a quality dimension that is relevant to consistency, as the extent to which the data elements from a data source are in agreement with the data elements from further individual data sources that report correlating effects.

Other quality dimensions

Many other data quality dimensions are relevant to the context of mobile crowdsensing environments. The authors of [Han et al., 2010] have characterized multidimensional requirements for the service of sensor data applications. The authors defined requirements for the quality of service in this context, such as reliability and timeliness of the services. They also defined requirements for the data from the sensors and focused on the accuracy and freshness of the data readings. Timeliness requirements are specified in the format of periodicity, deadline, or a certain relative order of different tasks [Han et al., 2010].

Timeliness is usually related to the age of the data and the degree of its validity in the system or in the real world [Serhani et al., 2016]. The authors of [Liu et al., 2019] defined timeliness for data collected by IoT devices as the extent to which an observation for the object is updated at a desired time of interest.

Currency was defined as the degree to which data is current or updated. Volatility was defined for sensor data as a value representing the variation of data over time [Han et al., 2010]. The authors of [Todoran et al., 2015] defined data currency as the percentage of extracted elements that are up-to-date. The authors of [Serhani et al., 2016] describe currency as a dimension that describes the extent to which data is up-to-date.

[Han et al., 2010] defined data availability as a dimension representing the accessibility of data for the intended use. Adequacy was also defined as an estimation of usability or quality of use.

[Liu et al., 2019] defined utility as the extent to which relevant data is accessed by data consumers from IoT datasets during a certain period of time.

2.3.2 . Data quality assessment

Numerous approaches have evaluated data quality for sensor data. This subsection discusses approaches that have proposed metrics to assess different data quality dimensions applicable to mobile crowdsensing environments.

Assessing data completeness

Many works focused on proposing metrics to evaluate data completeness in mobile crowdsensing environments. The work of [Serhani et al., 2016] proposed two metrics to evaluate data completeness. The first metric evaluates it as the ratio of the number of empty or null values over the total number of values as follows:

$$\text{completeness} = \frac{\text{number of empty values}}{\text{total number of values}}$$

The second metric for evaluating data completeness by the work of [Serhani et al., 2016] evaluates completeness as the total number of the stored actual records over the expected number of records as follows:

$$\text{completeness} = \frac{\text{Actual Total Number}}{\text{Expected Total Number}}$$

The authors of [Klein et al., 2007] propose a metric to measure the stream completeness c. The stream here represents the data stream of the measurements sent by a sensor, and the stream length is represented by m. Data stream completeness is defined as follows:

$$c = 1 - \frac{count(\text{missing Values})}{m}$$

The authors of [Rodríguez and Servigne, 2013] evaluated data completeness for sensor data in a mobile crowdsensing context by comparing the actual values with an estimated number of data records computed using the time period and the acquisition rate of the sensor.

The work of [Emran, 2015] surveyed many works assessing null-based, tuple-based, and schema-based completeness while the authors proposed an approach towards population-based completeness in which its completeness is determined by the number of missing individuals from a reference population. [Todoran et al., 2015] assessed data completeness for sensor data as the proportion of registered values over the expected values.

Completeness has also been addressed by the work of [Biswas et al., 2006], where the authors developed a quality model to assess data completeness for sensor data by translating data rates to completeness values measured over a period of time. They considered a specific *"smart home"* application context to demonstrate how completeness can be calculated. The authors define completeness in the

smart homes context, then define each system completeness, query completeness and present the metrics to evaluate both. The system completeness is defined as the fraction of actual measurements from the existing sensors over the expected measurements from these sensors:

$$SysComp = \frac{v}{n \cdot d}$$

Where v is the number of non-null values, n is the number of sensors in the system, and d is the duration of measurement of an application or a query. Then, to define query completeness, the authors first define the query data rate (QDR) and the system data rate ($SysDR$), where the $SysDR$ is the system's data rate. The $SysDR$ is defined as the maximum rate of all the resources in the system including the sensors. The QDR is the query data rate which is defined as the rate at which the application would like to see data delivered in a query response. It is dictated by the application requirements. Hence, the query completeness was defined as follows:

$$QComp = \frac{SysComp \cdot SysDR}{QDR}$$

where $SysComp$ is the system completeness, $SysDR$ is the system data rate, and finally, QDR is the query data rate.
Finally, the authors define the sensor data rate as the rate at which a sensor communicates data to the outside world. The authors consider that the rate at which a sensor outputs its data defines the completeness of a sensor. The sensor data completeness is defined as follows:

$$SysComp(s_i) = \frac{SensDR(s_i)}{SysDR}$$

where s_i is a sensor unit, $SensDR(s_i)$ is the output data rate of s_i, and $SysDR$ is the system's data rate.

The authors of [Dasu et al., 2016] address several data quality issues, among them missing and incomplete data. The authors propose data quality checks to assess completeness after the data gathering process. One of these checks is the proportion of received data files over the expected number of data files. Another data quality check relevant to data completeness is the proportion of the size of the received data file over the expected file size. The expected size of each data file indicates whether the file is complete or not. This means that if the size of the actual data file is the same as that of the expected file, then they are complete, and no further checks on the content of the files are necessary.

Assessing data accuracy

Other data quality dimensions have been assessed in mobile crowdsensing environments, such as data accuracy. The authors of [Serhani et al., 2016] evaluated $data\ accuracy$ for health data collected using sensors on patients. $Accuracy$ was

evaluated as the ratio between the number of correct values stored and the total number of values as follows:

$$\text{accuracy} = \frac{\text{Number Of Correct Values}}{\text{Total number of Values}}$$

Data accuracy was evaluated by [Rodríguez and Servigne, 2013] compared to a reference value from a reference datase, and while considering the sensor technical precision by the manufacturer. The work of [Todoran et al., 2015] proposed to measure data accuracy as the rate between the correct values and the total number of values.

The authors of [Fizza et al., 2022] suggest that data accuracy is measured based on the stability of sensor data. They compute the variation of data value v, coming from a sensor, relative to its mean using moving standard deviation (mSD) as follows:

$$mSD_k = \sqrt{\frac{1}{m-1} \sum_{i=k-m+1}^{k} (v_i - \bar{v})^2}, \ \forall k = 1, \ldots, n$$

Where v_i is the data value at the i$^{\text{th}}$ row, $\bar{v}$ average of data values, n is the number of rows, and m is the time window size defined by the application.
Then the accuracy of sensor data is defined by this work as follows:

$$\text{A} = \left(1 - \frac{\sum_{i=1}^{k} \frac{mSD_i}{\sqrt{m}}}{n/m}\right)$$

Where n is the number of rows and m is the time window size defined by the application.
In [Karagulian et al., 2019], the authors have reviewed the existing works related to these types of sensors and they have introduced a comparison of the existing studies and an evaluation of the agreement between low-cost sensors and reference datasets.

Assessing data consistency

[Dasu et al., 2016] proposed a framework to measure data quality for temporal streams. The authors tackle data quality issues in temporal data and address more specifically spikes in the data or some inconsistencies that can deteriorate the value of data quality dimensions such as the consistency and the accuracy of the data. The approach aims at detecting data glitches or errors in the data. This is done by comparing the data to some constraints defined on the data using a statistical distortion approach that measures the distance between a defined reference, and the actual data. The authors apply some quality checks on the content of the data to assess the accuracy and consistency of the data. These quality checks include the definition of different types of data constraints. For each type of

data constraint, the approach discusses using statistical distortion to compute the distance between the ideal, which is the reference dataset, and the actual data.

Consistency was also evaluated by the authors of [Serhani et al., 2016] as the ratio of the total number of inconsistent values over the total number of values as follows:

$$\text{consistency} = \frac{\text{Number of Inconsistent Values}}{\text{Total number of Values}}$$

Assessing data integrity

The author of [Ray, 2018] first introduces the context of maritime vessels on spatial data and how everything works within an *Automatic Identification System* (AIS). Then, the author highlights the weaknesses in such systems in terms of message falsification attacks or spoofing by external malicious actors. A variety of data types are introduced in the context as follows: the navigation data indicating the positions of the vessels as acquired by the AIS receivers, the vessel-oriented data indicating the official nominative vessel position, the geographic data (cartographic, topographic), and finally, the environmental data such as the weather and ocean conditions data from forecast models and actual observations. The work assesses the integrity of maritime messages through message-based and signal-based analysis. The message-based approach relies on four ways to identify the integrity of the messages. The four ways use a comparison of the individual fields within a message and the message type to infer the integrity of a message. As for the signal-based analysis, the author considered several parameters such as the power of the received signal and others that are time-dependent and relative to the shape of the signal.

Assessing micro-sensing units

Some works have also addressed quality evaluation at the sensor level such as [Fishbain et al., 2017] who proposed a toolkit for the evaluation of micro-sensing units explaining all the factors and their metrics. The toolkit consisted of eight different measures for the quantification of the quality of a sensor unit compared to data from a reference device as follows:

- **RMSE and Pearson correlation**: RMSE measures the total bias (deviation) between two time series, while Pearson evaluates the correlation between two time series.

- **Kendall and Spearman**: both are correlations that are sensitive to monotonic but non-linear relationships because the correlation between a reference and a sensor does not necessarily have to be linear, which is an aspect that RMSE and Pearson cannot deal with.

- **Presence**: this measure accounts for the sensor's availability of a measurement at a given time where limited presence always brings the question of representativity of the measurements.

- **Source Analysis**: this measure assesses the ability of the sensor device to react to changes in observations within a time interval that corresponds to wind direction and be sensitive to concentration changes by calculating bivariate polar plots (Pearson Correlation Coefficients) between the reference dataset and the data from the sensors treating them as two-dimensional matrices.

- **Match score**: it is the proportion of agreement among strata of partitions between a reference dataset and the sensor measurements. Both the reference dataset and the data coming from the sensors are divided into d partitions. Then the partition labels of all the measurements at each studied timestamp are compared among the reference and the sensor data to compute the agreement between the two datasets.

- **lower frequencies energy (LFE)**: is rather a characteristic of the sensor than a metric to measure the quality of the data. The signal's energy in the lower frequencies can be used for evaluating the capability of the sensor to capture the temporal variability of the pollutant. Thus, the smaller the energy portion in the higher frequencies, the better the sensor can capture the signal's temporal variability

2.3.3 . Analysis

We have studied definitions and metrics to evaluate several data quality dimensions relevant to mobile crowdsensing environments. We have presented metrics for completeness, accuracy, consistency, and integrity.

For data accuracy, most definitions in the existing works have defined data accuracy as the dimension of data that measures how close a measurement is to describing a real-world phenomenon. Data accuracy was either evaluated with the closeness of a value to a reference value or with the ratio of correct values over the total.

Data completeness in the context of mobile crowdsensing environments has also been associated with the presence of data and with missing values. The difference with the definitions in section 2.2 is that the missing values are not just identified with null values but also with entire records missing.
There are two metrics proposed to evaluate data completeness. The first is the proportion of missing values from the total number of values. And the second measures the available number of data values compared to some expected value. Some works defined this expected value using the sampling rate of the measuring sensors.

In order to assess data consistency, we need some prior knowledge about

the data in the form of rules or data constraints, which are not always easy to define.

These studied approaches fail to measure the quality dimensions considering the different aspects of the data being studied, such as the time, location, or the sensor unit taking the measurements. Consider that the data completeness quality dimension is being studied for sensor data in an air pollution context where the sensors are carried by users at different timestamps and locations. Evaluating data completeness using some existing metrics from the literature would result in a value that relies on the frequency at which a sensor is supposed to take measurements. The only aspect of the data that is considered is the sensor. Other aspects of the data, such as the user, the time, and the location, are not considered. The studied existing metrics fail to capture in this example is, for example, the completeness of the data of a specific location, or of a certain period of time, or even the completeness of the user. The existing metrics of completeness for instance, need to consider the completeness of the different aspects of the data that characterize mobile crowdsensing environments, not just focus on the sensor as done in the work of [Biswas et al., 2006].

To the best of our knowledge, the existing metrics fail to answer these questions. Therefore, new metrics considering the different aspects of sensor data need to be defined. Some works such as [Östman, 1997] have considered defining accuracy from several aspects: thematic, temporal, and positional for spatial data. However, no works have addressed the sensor data in a mobile crowdsensing context, and no works also focused on the different understandings of data completeness in this context.

2.4 . Data imputation for completeness improvement in MCS

Missing data is a major problem specifically in mobile crowdsensing environments. To improve the completeness of the data, data imputation and other techniques are used to replace the missing values. Data imputation techniques and inference have gained massive popularity in research due to the impact of missing points or chunks of data on the quality of the indicators resulting from data analysis. A wide variety of approaches have been proposed to generate missing values to improve the quality of the data. These approaches aim to avoid faulty indicators and analysis since poor-quality data can result in poor-quality analysis. The approach presented in [Wang et al., 2016] infers data of unsensed cells in a mobile crowdsensing environment for meteorological and traffic data using k nearest neighbors (KNN), compressive sensing (CS), and spatio-temporal compressive sensing (STCS). Several families of data imputation techniques exist to address different types of data and different application domains [Khayati et al., 2020]. The imputation techniques could be categorized into 3 different families according to [Khayati et al., 2020]:

- *Matrix-based approaches* such as SVDImpute [Troyanskaya et al., 2001], SoftImpute [Mazumder et al., 2010], CDRec, that will be discussed in fur-

ther details later on in this section. This family of techniques transforms the data using dimensionality reduction methods. Principal Component Analysis (PCA) [Jolliffe, 2011] and Singular Value Decomposition (SVD) [Skillicorn, 2007] are the most commonly used techniques for dimensionality reduction. Other techniques exist such as Centroid Decomposition (CD) [Chu and Funderlic, 2002], Matrix Factorization (MF) [Koren et al., 2009], and Non-Negative Matrix Factorization (NMF) [Kim et al., 2015].

- *Pattern-based approaches* such as DynaMMo [Li et al., 2009], ST-MVL [Yi et al., 2016]and TKCM. This family of approaches rely on finding high similarity patterns in the data in order to impute a missing value. They study the pattern of the missing block and then look for candidate replacement patterns in other series.

- *Neural networks-based* like LSTM and other recurrent neural networks (RNN) techniques. This family employs neural networks to study and generate the missing blocks of data.

In the following subsections, we present approaches from the first two families of approaches: pattern-based data imputation techniques and matrix-based data imputation techniques. We describe some approaches from these two families that address missing data challenges within the context of mobile crowdsensing environments.

2.4.1 . Matrix-based Techniques

Matrix-based data imputation techniques work by leveraging the patterns and relationships present in the available data to estimate and fill in the missing values within a matrix structure. A common approach in this family is to use matrix factorization methods, such as singular value decomposition (SVD) or principal component analysis (PCA), to decompose the data matrix into lower-dimensional representations.

The authors of [Troyanskaya et al., 2001] propose two techniques to estimate missing values for DNA microarrays data: SVDImpute and KNNImpute. SVDImpute employs the *Singular Value Decomposition* (SVD) to obtain the principle components of the matrix containing the gene expression microarrays in every row in the matrix. For datasets with missing values, the first step is replacing missing values in the data matrix A by row average because SVD works only on complete matrices. Then, SVD factorizes the matrix A, containing the data, into 3 singular matrices as follows:

$$A_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n}$$

Matrix $V^T_{n\times n}$ contains eigenvectors that are quantified by their corresponding eigenvalues on the diagonal of matrix Σ, and U contains the left singular vectors. The eigenvectors represent gene expresssion microarrays, and are also referred to as eigengenes. After the principle components are computed, the k most significant eigengenes are selected. Then, we estimate a missing value j in gene i by first

regressing this gene against the k eigengenes to get the coefficients $(\beta_0, \beta_1, ..., \beta_n)$ in the below regression equation:

$$\hat{v}_{ij} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon \tag{2.1}$$

Where β_0 is the intercept, $(\beta_1, ..., \beta_n)$ are the regression coefficients and ϵ represents the residuals. The approach finally uses the coefficients of the regression to reconstruct the missing measurement j from a linear combination of the k eigenvalues. All missing values in the data matrix A are imputed using the technique aforementioned to provide a new matrix $A\prime$. Finally, the difference between A and $A\prime$ is computed, and if the difference is greater than a certain threshold, then a new iteration is started and all the steps mentioned earlier are repeated. In the new iteration, the value of A is set to $A\prime$ and the same steps computing the new $A\prime$ are repeated to compute the new $A\prime$ until it converges. SVDImpute in this case is iterative SVD.

The authors of [Troyanskaya et al., 2001] also presented another approach for data imputation: $KNNImpute$. This approach is based on k-nearest neighbors applied on the data matrix containing data measurements of *k* sensors at timestamp *t*. KNNImpute generates a missing value at timestamp t based on the measurements from neighboring sensors with the same timestamp. The value of the measurement from each sensor is weighted according to its similarity with the target sensor s_t. The similarity metric is the Euclidean distance between a sensor s_i and the target sensor s_t with the missing measurement. Finally, a weighted average of the k-nearest sensors is computed to impute a missing value using similarity as the weight. The missing value inferred using KNNImpute is defined by:

$$\hat{v}_j = \frac{\sum_{l=1}^{k} v_{lj} * d_{s_l,s_t}}{\sum_{l=1}^{k} d_{s_l,s_t}} \tag{2.2}$$

Where $\hat{v}_j$ is the imputed value of target sensor s_t at timestamp t_j, *k* is the number of nearest sensors, v_{lj} is the measurement value of sensor s_l at timestamp t_j, and d_{s_l,s_t} is the Euclidean distance between sensor s_l and target sensor s_t.

Computing distances between data points without taking the quality of the measurements has a negative impact and can deteriorate the imputation precision. For instance, two measurements can be geographically close but in different environments, as one could be captured inside a bus and the other in the street, which could have a great impact on the imputation. Techniques that rely on the study of correlations between the data points without considering the quality of the measurements could end up finding correlations between points with actual high values and others that are just spikes or noise due to a quality issue, which will result in a faulty imputation.

2.4.2 . Pattern-based Techniques

Pattern-based data imputation techniques are approaches used to fill missing values in datasets by identifying patterns and relationships within the available data. When dealing with datasets with missing values, imputation methods aim to estimate or predict the missing values based on the patterns observed in the existing data. In this subsection, we explore some pattern-based data imputation approaches that were proposed for time series data.

The DynaMMo Approach

The authors of [Li et al., 2009] developed a pattern-based approach relying on expectation maximization to identify hidden variables in order to recover missing values. This approach leverages the two main characteristics of time series: *temporal continuity* and *spatial correlation*.

Given a time series X with duration T (T measurements) in m dimensions, $X = \{x_1, \ldots, x_T\}$, W is a matrix indicating missing values where $w_{t,k} = 0$ indicates that the k-th dimensional observation is missing at time t and $w_{t,k} = 1$ means the observation is present, the observed part is denoted by X_g, and the missing part as X_m. The authors use expectation maximization to estimate the missing values conditioned on the observed ones $\mathbb{E}[X_m|X_g]$.

However, since it is difficult to directly maximize the data likelihood in the missing value setting, the authors maximize the expected log-likelihood of the observation sequence instead. They use a sequence of latent variables z_n, to model the hidden patterns of the observation sequence. The estimated sequence $\hat{X}$ is then computed.

First, the sequence $\hat{X}$ is initialized with the data from the observed (actual) sequence X_g, and the missing values are filled using interpolation methods. The authors assume a mapping function **G** that captures the correlations between the observation dimensions. These learned correlations will help infer missing dimensions from the latent variables. The mapping **G** is a linear projection matrix from the latent variables to the data sequence (both observed and missing) for each time tick. The authors leverage the temporal continuity in time series and assume that the latent variables are time-dependent on the values determined from the previous timestamp. Temporal continuity in time series refers to the sequential and uninterrupted flow of data points over time, capturing possible dependencies between successive observations. This temporal continuity is modeled by a linear mapping F.

Where z_0 is the initial state of the latent variables. **F** is the linear mapping between the latent variables and the values determined from the previous timestamp, and **G** is the observation projection. w_0, w_i i (i = 1 ... T), and ϵ_i are multivariate noises with Gaussian distributions.

Then, the algorithm iteratively estimates the latent variables, maximizes with

respect to parameters and estimates the missing values until convergence. The goal of estimating the parameters $\theta = [\mathbf{F}, \mathbf{G}, \mathsf{z}_0, \Gamma, \Lambda, \Sigma]$ is achieved through maximizing the likelihood of observed data, $\mathcal{L}(\theta) = \mathsf{P}(\mathsf{X}_g)$. But since it is complicated to directly estimate the data likelihood, the authors chose to maximize the expected log-likelihood of the observation sequence. Then, the authors define the objective function as the expected log-likelihood $\mathsf{Q}(\theta)$ with respect to the parameters $\theta = [\mathbf{F}, \mathbf{G}, \mathsf{z}_0, \Gamma, \Lambda, \Sigma]$:

$$Q(\theta) = \mathbb{E}_{X_m, Z|X_g, W}[P(X_g, X_m, Z)]$$
$$= \mathbb{E}_{X_m, Z|X_g, W}[-D(\mathbf{z}_1, z_0, \Gamma) - \sum_{t=2}^{T} D(\mathbf{z}_t, \mathbf{F}\mathbf{z}_{t-1}, \Gamma) - \sum_{t=1}^{T} D(\mathbf{x}_t, \mathbf{G}\mathbf{z}_t, \Sigma) - \frac{\log|\Gamma|}{2}$$
$$- \frac{(T-1)\log|\Lambda|}{2} - \frac{T\log|\Sigma|}{2}]$$

Where W is the missing value indication matrix, $D()$ is the square of the Mahalanobis distance $D(x, y, \Sigma) = (x - y)^T \Sigma^{-1} (x - y)$.

Finally, to estimate the missing values, the authors use a belief propagation algorithm to estimate the posterior expectations of latent variables using the Markov property, so they propose to estimate a missing value $X_{i,j}$ as follows:

$$\hat{X}_{i,j}^{new} = G^{new} . \mathbb{E}[Z|\hat{X}; \theta]_{\{i,j\}}$$

Where the authors later take the derivatives of the objective function equations with respect to the parameters of θ^{new} and set them to zero in order to estimate the new parameters.

The ST-MVL Approach

The approach presented in [Yi et al., 2016] is also a pattern-based approach that uses a multi-view learning algorithm to compute a weighted sum of imputed values from four different algorithms on both local and global views. The global view represents the whole dataset, and the local view represents a local data matrix constructed using measurements in the spatial and temporal neighborhood of the missing measurement. On the global view, the *Inverse Distance Weighting* (IDW) technique is used to impute missing value based on spatial aspects while *Simple Exponential Smoothing* (SES) generates a missing value based on temporal aspects. On the local view, *User-based Collaborative Filtering* (UCF) is an imputation technique that generates a missing value based on spatial distance in a defined window size and *Item-based Collaborative Filtering* (ICF) imputes missing values based on temporally adjacent measurements in a defined time window size. Multi-view learning of the weighted sum of all the aforementioned techniques represents the final generated value by ST-MVL.

The first sub-technique of ST-MVL is the *Inverse Distance Weighting* (IDW), which is a statistical model used to interpolate missing values at a given timestamp, based on the spatially closest sensor measurements. To estimate a missing measurement $\hat{v}_{gs}$, IDW retrieves all measurements from geospatially adjacent sensors at timestamp t and assigns weight to every measurement according to its distance from the target sensor which is the sensor with the missing measurement. This weight assignment according to the distance gives the closest sensors a higher weight than further ones. The measurements are then aggregated using the following equation to generate the missing value $\hat{v}_{gs}$:

$$\hat{v}_{gs} = \frac{\sum_{i=1}^{m} v_i * d_i^{-\alpha}}{\sum_{i=1}^{m} d_i^{-\alpha}} \tag{2.3}$$

Where d_i is the spatial distance between sensor i and target sensor. α is a positive power parameter.

α controls the decay rate of sensor's weight by $d_i^{-\alpha}$. A higher α means a faster decay of weight by distance.

The second sub-technique is *Simple Exponential Smoothing* (SES), which is a technique used to estimate a missing measurement based on the most recent adjacent measurements from the same sensor. To estimate a missing measurement $\hat{v}_{gt}$, SES takes all measurements from the measuring sensor and assigns a weight to a measurement according to its time-adjacency to the missing measurement. The measurements are later aggregated according to the following equation:

$$\hat{v}_{gt} = \frac{\sum_{j=1}^{n} v_j * \beta * (1-\beta)^{t_{j-1}}}{\sum_{j=1}^{n} \beta * (1-\beta)^{t_{j-1}}} \tag{2.4}$$

Where t_j is a time interval between a candidate measurement v_j and target measurement. β is a smoothing parameter that ranges between 0 and 1.

β controls the decay rate of weight over time intervals as temporally closer measurements get higher weights. $\beta * (1-\beta)^{t-1}$ indicates the weight given to a measurement, and assigns a higher weight to temporally closer measurements than the distant ones.

The third sub-technique is *User-based Collaborative Filtering* (UCF), which is an algorithm used to impute a missing measurement at a given timestamp by computing the similarity between sensor measurements and a target sensor for a given window size w. To estimate a missing measurement $\hat{v}_{2j}$, UCF first constructs a local data matrix which is a subset of the existing data limited by the window size w. The target sensor is the sensor that has the missing measurement in study.

It then computes the similarity between every sensor s_i and the target sensor s_1 for every measurement in the local data matrix as follows.

$$sim(si, s_1) = \frac{1}{\sqrt{\frac{\sum_{k=j-\frac{w-1}{2}}^{j+\frac{w-1}{2}} (v_{ik}-v_{1k})^2}{NT}}} \tag{2.5}$$

Finally, a weighted average $\hat{v}_{ls}$ is computed where the weight is the similarity score as follows.

$$\hat{v}_{ls} = \frac{\sum_{i=1}^{m} v_i * sim_i}{\sum_{i=1}^{m} sim_i} \tag{2.6}$$

NT is the number of timestamps where both sensors have measurements. w is the window size. The local data matrix is a subset of the data where a row stands for a sensor and a column denotes a timestamp. It is built from data from all the sensors in the pre-defined spatial neighborhood, and from the time-adjacent measurements according to the window size w, and the timestamp j of the missing measurement. The measurements $v_{i,j}$ from all sensors in the spatial neighborhood are included in the data matrix and denoted by $v_{*,j}$. The local data matrix is constructed as follows: $[v_{*,(j-(w-1)/2)}, \ldots, v_{*,(j+(w-1)/2)}]$.

The last sub-technique is *Item-based Collaborative Filtering* (ICF), which is an algorithm used to impute a missing measurement by computing the similarity between temporally adjacent measurements within a preset window size w. To estimate a missing value, this technique constructs a local data matrix which is a subset of the initial dataset. The similarity between two timestamps based on measurements from the local data matrix is then computed:

$$sim(t_1, t_2) = \frac{1}{\sqrt{\frac{\sum_{i=1}^{m} (v_{i1}-v_{i2})^2}{NS}}} \tag{2.7}$$

Where m is the number of available sensors. A weighted average $\hat{v}_{lt}$ where the weight is the similarity score is then computed to estimate a missing value as follows:

$$\hat{v}_{lt} = \frac{\sum_{j=j_1}^{j_2} v_j * sim_j}{\sum_{j=j_1}^{j_2} sim_j} \tag{2.8}$$

Where NS is the number of sensors that have measurements at both timestamps t_1 and t_2, and w is the window size.

Finally, ST-MVL integrates the predictions of the four aforementioned sub-techniques to generate a final result using a multi-view learning algorithm that

computes a weighted sum of the generated predictions according to the following formula, where the weights of each technique are learned during the training phase:

$$\hat{v}_{mvl} = w_1 * \hat{v}_{gs} + w_2 * \hat{v}_{gt} + w_3 * \hat{v}_{ls} + w_4 * \hat{v}_{lt} + b$$

where b is a residual and $w_i (i = 1, 2, 3, 4)$ is a weight assigned to each view.

Analysis

Matrix-based approaches use dimensionality reduction techniques such as singular value decomposition (SVD) or *Principal Component Analysis* (PCA) to reduce the dimension of the feature set and expose the number of linearly independent dimensions and the linear relationships in the data. Such approaches are also called matrix completion (recovery) approaches because the data is represented in a data matrix where the algorithm recovers the missing data measurements. Some dimensionality reduction techniques such as SVD and PCA rely on linear relationships between the data features but these relationships do not always necessarily exist depending on the data. This family of approaches works better for high-dimensional data than pattern-based approaches because it has the advantage of reducing the dimension space to a size that is convenient for the generation of missing values using data mining techniques. Matrix-based approaches could also work better on large datasets because the dimensionality reduction techniques could reduce the data size. The pattern-based family of techniques might struggle with the curse of dimensionality and the computation time of patterns in large datasets.

Pattern-based approaches learn patterns in the data in order to estimate the missing values of the data. This family of approaches works best on datasets with high similarities in the data series. When a block is missing in some series, the algorithm would use the similarity to several other series in order to generate the missing values in the block. This family of techniques requires adequate parameterization. For example, one of the parameters is the size of the patterns we are looking for. If this parameter has a very low value, we might miss real pattern anomalies in the data that have a greater length. Moreover, if this parameter is too high, the similarity computation becomes costly in terms of computational time. Data imputation using this family of approaches on large datasets can also be expensive in terms of computational time. This family of approaches works best on data with high correlations among the time series, which means that the values of the measurements in the different series vary closely together. We consider as an example an air quality monitoring context with several sensors measuring different pollutants. If a sensor loses data, pattern-based approaches can be very useful since there are many air pollutants that are highly correlated to each other.

Another type of pattern-based approaches are the works that manually set similarity rules which sometimes can be very accurate because of human

validation. However, these works require an important amount of knowledge on the data and domain experts in order to be able to set these similarity rules. This can be expensive in terms of time, cost, and resources.

One of the characteristics of both families of data imputation approaches is that they rely on available measurements to generate a missing value, either using measurements from the same sensor at different timestamps or from other sensors. However, these data measurements could be coming from sensors facing quality problems such as calibration issues, inaccurate measurements, anomalies, etc. This means that the imputations will rely on data of poor quality, which implies that the quality of the imputed values will also be poor. One way to improve these approaches, given the quality of the measuring sensors, would be to take into account this quality during the data imputation process. This means that measurements from high-quality sensors will be given higher priority than those coming from lower-quality sensors. This will lead to a greater impact of measurements with high quality than the ones with poor quality, and therefore to more reliable indicators computed from this data.

2.5 . Anomaly Detection in Time Series

Anomaly detection has long been a research problem in large databases and in wireless sensor networks. Mobile, low-cost sensors are subject to many points of failure, spikes in acquired measurements, and malfunctions. Sometimes, meteorological events such as tornadoes, heatwaves, wildfires, and sandstorms are represented as abrupt spikes and unusual patterns in the data called pattern anomalies. We are interested in detecting pattern anomalies. Sometimes, an anomaly is an inconsistent data point that is either a spike or a noise and hence has to be removed because it disrupts the quality of the data. At other times, an anomaly is a characterization of an unusual data sequence, also called an unusual pattern, that can provide many insights about the data once detected. These patterns can reveal an interesting new behavior or important events that need to be addressed.

This section introduces anomalies and focuses on the existing definitions and types of anomalies found in mobile crowdsensing environments. It also presents studied works from the different families of anomaly detection approaches: statistical-based, deviation-based, clustering-based, and distance-based approaches. We conclude this section by analyzing the studied works.

2.5.1 . Existing definitions and types of anomalies

The problem of detecting and correcting outliers is not new [Grubbs, 1950, Grubbs, 1969]. Before we discuss the different types of anomalies in the literature, we start with the meaning of this notion of outliers and the existing definitions of outliers/anomalies. These definitions were proposed in different contexts such as traditional databases and time series data.

It should be noted that there is no universally accepted definition of the notion of outliers. Several works in different domains have addressed outlier detection. For example, in [Hawkins, 1980], an outlier is defined as an observation that deviates so much from other observations as to arouse suspicions that a different mechanism generated it. In [Aggarwal, 2016], outliers are referred to as abnormalities, discordants, deviants, or anomalies in the data mining and statistics literature. [Grubbs, 1969] defined an outlying observation, or outlier, as one that appears to deviate markedly from other members of the sample in which it occurs. In [Gupta et al., 2014], outliers are considered deviations from expected values (forecasts). In [Braei and Wagner, 2020], the authors defined an anomaly as an observation or a sequence of observations that deviates remarkably from the general distribution of data. The set of anomalies forms a very small part of the dataset. The authors of [Chandola et al., 2009] also defined anomalies as patterns in data that do not conform to a well-defined notion of normal behavior.
To summarize, outliers have been defined as deviations in some data observations that arouse suspicion. This definition was later extended in mobile crowdsensing environments to deviation in a data observation or a sequence of data observations from the expected values, forecasts, or distributions.

Existing types of anomalies

Many works have defined types of anomalies in different contexts. [Blázquez-García et al., 2021] differentiate between the meaning of anomalies that are unwanted data and have to be detected then cleaned, and anomalies that are events of interest and hence, need to be detected and studied in time series data. The authors also defined three types of outliers: a *point*, a *subsequence*, and a *series*.

- **Point outlier**: A point outlier is a datum that behaves unusually at a specific point in time when compared either to the other values in the time series where it is considered a global outlier or compared to its neighboring points where it is considered a local outlier.

- **Subsequence outlier**: This term refers to consecutive points in time whose joint behavior is unusual, although each observation individually is not necessarily a point outlier.

- **Outlier time series**: Entire time series can also be outliers, but they can only be detected when the input data is a multivariate time series such that the other variables that have normal values can help identify the outliers in one variable.

There are other typologies of anomalies in the literature. For example, the authors of [Braei and Wagner, 2020] define the following three types of anomalies in time series.

- **Point anomalies**: If a point deviates significantly from the rest of the data, it is considered a point anomaly. For example, a big purchase transaction that significantly differs from other transactions is a point anomaly. Hence, a point X_t is considered a point anomaly, if its value differs significantly from all the points in the interval $[X_{t-k}, X_{t+k}]$, k $\in$ R and k is sufficiently large.

- **Collective anomalies**: There are cases where individual points are not anomalous, but a sequence of points is labeled as an anomaly. For example, if a bank customer withdraws \$500 from her bank account every day of the week. Although withdrawing \$500 occasionally is normal for the customer, this sequence of withdrawals is an anomalous behavior.

- **Contextual anomalies**: Some points can be normal in a certain context, while detected as an anomaly in another context. For example, in a meteorological context, having a daily temperature of 40°C in summer is normal, while the same temperature during the winter is regarded as an anomaly.

Detecting unusual pattern anomalies is far more challenging than detecting point anomalies [Gupta et al., 2014] because they are less trivial to detect, and there are more parameters that need to be considered to detect them. For example, the size of the pattern to look for in the data could vary depending on the data and the context, knowing which patterns are normal and which are not, requires more knowledge about the data than a single point whose value is outside the accepted range of the measured phenomena. The work of [Gupta et al., 2014] studied outliers within a given time series. The authors tackled two distinct types of outliers: the first is single data points, which are referred to as point outliers. The second is a sequence of several consecutive data points in a series that are denoted by subsequence outliers.

2.5.2 . Statistical-based methods

The statistical techniques are methods that use statistical concepts to detect an anomaly. Several works have used statistical-based methods for anomaly/outlier detection. The authors of [Bakar et al., 2006] compared the performance of three outlier detection techniques on air quality data, among which is the $control\ chart\ technique$ (CCT).

The CCT technique [Hackl and Ledolter, 1991] is a statistical-based method usually used to determine whether a process is operating in statistical control or not. It detects any unwanted changes in the process. It consists of a center line that is the average of all samples plotted. Upper and lower control limits define the constraints of common variations out of which any data point will be considered as an outlier. The center line is plotted over time as follows.
The average of the samples plotted is computed by:

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

where X_i is a data measurement value (X_i,...,X_n), n is the total number of the data.
Upper (UCL) and lower (LCL) limits are computed by:

$$UCL = \bar{X} + Z\sigma_x$$
$$LCL = \bar{X} - Z\sigma_x$$
$$\sigma_x = \frac{\sigma}{\sqrt{n}}$$
$$\sigma = standarddeviation = \left(\frac{\left(X_i - \bar{X}\right)^2}{n-1} \right)^{1/2}$$

The authors of [Zhang et al., 2020] employs the median filter (MF) statistical method as a preprocessor to detect obvious anomalies. The median filter (MF) [Brownrigg, 1984] preprocessor is a window-based statistical algorithm that is useful in reducing random noise. The median filter operates on a sliding window of a predefined length. Within each window, the median value is calculated and compared with a predefined threshold. If the difference between the current data point and the median value exceeds the threshold, it is considered an anomaly.

A data quality assessment framework, called *Hygieia*, has been proposed in the work of [Aquino et al., 2019] within a smart sensor network context to detect outliers. The authors defined a framework that receives a continuous flow of data and then computes the interquartile range (IQR) of the data, according to which they evaluate several quality aspects of the data. Hygieia sorts the data and calculates the 25 (P25) and 75 (P75) percentiles. It then analyzes if the received data is greater than the upper boundary or smaller than the lower boundary. If the received data is lower or greater than the boundaries, it is considered an outlier.

The approach presented in [Giannoni et al., 2018] experiments the performance of several techniques, two of which are statistical-based: the *average low-high pass filter* and the *seasonal ESD algorithm*.

The *average low-high pass filter* is a widely used statistical-based solution in the field of sensors' anomaly detection. The basic idea is to make use of the running average computed based on the last W acquisitions, W being a sliding window of fixed size. A data point will be detected as anomalous if it significantly differs from the running average. There are two implemented versions of this technique, the online and the offline version. The offline version assumes the entire series is available and therefore has the standard deviation of it. Hence, for a new data point coming, if the distance from this point and the running average is greater than the standard deviation, this point is considered anomalous. The online version is similar, only the standard deviation is not computed for the whole series but approximated for the acquisitions in the window at each iteration.

The *Seasonal-Extreme Studentized Deviate algorithm (S-ESD)* is also a

statistical-based algorithm released by Twitter in [Hochenbaum et al., 2017], and can only be applied on time series with the symmetrically distributed residual component. First, the seasonal S and the trend T components are extracted. The median X^*, which can be regarded as a stable approximation of the trend, is computed from the trend. Then the residual is computed as R = X - S - X^*. The algorithm then uses the ESD statistical test to identify any anomalous events.

2.5.3 . Deviation-based methods

Deviation-based methods use past measurements or data to train a model to predict an upcoming measurements in the data. The underlying idea of these methods is that if the predicted value deviates significantly from the observed value, the observed value is regarded as anomalous. There are several prediction based methods, some are simple with no or little parameters to define and tune such as ARIMA models and the 1-class SVMs, while others that are more complicated to train, find the proper parameters like RNNs and LSTMs.

The authors of [Bakar et al., 2006] studied an anomaly detection method using linear regression. The linear regression technique [Aalen, 1989] is used to evaluate the strength of a relationship between two variables, a dependent and an independent one. The linear regression technique estimates the linear relationship between x, the predictor, and y, the response variable as follows:

$$y = \alpha + \beta x$$

where the variance of y is assumed constant and α and β are regression coefficients specifying the y-intercept and the slope of the line respectively.
Given s data points of the form (x_1, y_1), (x_2, y_2), ... , (x_s, y_s), α and β are estimated as follows:

$$\beta = \frac{\sum_{i=1}^{s} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s} (x_i - \bar{x})^2}$$

$$\alpha = \bar{y} - \beta \bar{x}$$

where $\bar{x}$ is the average of x_1, x_2, ... , x_s and $\bar{y}$ is the average of y_1, y_2, ... , y_s.

The survey conducted by [Braei and Wagner, 2020] introduces a set of deviation-based methods. The auto-regressive (AR), moving average (MA), auto-regressive moving average (ARMA), and auto-regressive integrated moving average (ARIMA) models are statistical techniques that estimate a measurement value and then compute its deviation from the actual value to identify whether it is anomalous or not.

Auto-regressive model is a basic method for univariate time series. It is a linear model where X_t, the dependent variable, is based on a finite set of previous

values having length p known as independent variables, and an error value ϵ. A measurement is estimated using this model as follows.

$$X_t = \sum_{i=1}^{p} a_i.X_{t-i} + c + \epsilon_t$$

The auto-regressive model, in this case, is of order p and is represented by AR_p. The error values ϵ_t are considered uncorrelated and have a constant mean of zero and constant variance σ. To detect an outlier, we compute an outlier score which is the difference between the estimated value and the observed/actual one. AR models assume that the data is stationary, so in case the data is not, it has to be transformed.

The moving average model considers the current observation X_t to be a combination of the last q prediction errors ϵ_t,ϵ_{t-1},...,ϵ_{t-q}. The MA model equation estimates is computed as follows.

$$X_t = \sum_{i=1}^{p} a_i.X_{t-i} + \sum_{i=1}^{q} b_i.\epsilon_{t-i} + \epsilon_t$$

The coefficients a_0,...,a_q are learned from the data. Unlike in the auto-regressive model, the errors in the moving average model are known after the model is fitted.

A time series of the ARMA(p, q) model is dependent of the last p observations and q errors:

$$X_t = \sum_{i=1}^{q} a_i.\epsilon_{t-i} + \mu + \epsilon_t$$

X_t is a an ARMA model iff X_t is stationary, μ is the mean of the data. The main challenge here is to select the proper p and q because high values of p and q can result in overfitting the model and hence, a high number of false negatives in anomaly detection. In case p and q are too low, this leads to underfitting the model and hence a high number of false positives. The ARIMA model is a more generalized version of the ARMA model. The data can be non-stationary and in addition to the p and q parameters, there is the d parameter that states how many times the series is differenced.
For $d = 1$, the time series x_0,...,x_T is differenced as follows:

$$X'_t = X_i - X_{i-1} \forall i \in 1, ..., T$$

The authors of [Zhang et al., 2020] present a deviation-based method used to predict the value of the tested data point based on two stacked LSTM cells. The control chart technique is then used to compute the deviation of the actual data measurement from the accepted range of values computed by this technique. The stacked model consists of multiple LSTM layers. It improves the training efficiency and obtains higher accuracy by adding depth to the network. One LSTM cell

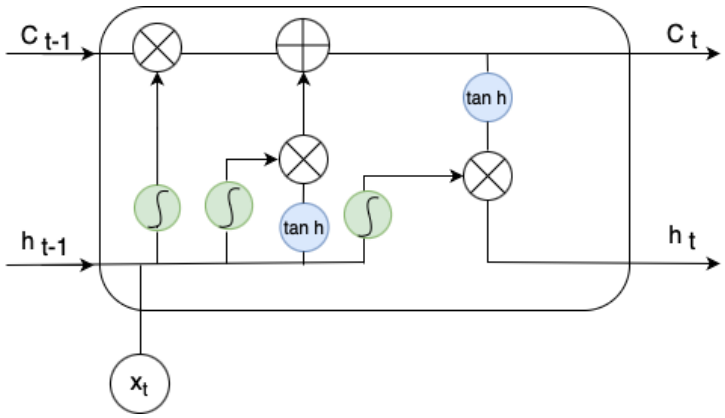

Figure 2.2: Internal architecture of an LSTM cell by [Zhang et al., 2020].

is shown in Figure 2.2. The prediction of output h_t of a single LSTM layer is calculated as follows:

$$h_t = o_t * tanh(c_t)$$

Where o_t is the output gate, and c_t is the current cell. These two gates are computed as follows:

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)$$

Where W_o represents the weight matrix, and b_o is the bias. The current cell c_t is computed as follows:

$$c_t = f_t * c_{t-1} + i_t * c_t$$

Where "*" represents the dot product, c_t represents the current cell, and c_{t-1} represents the last cell.

[Giannoni et al., 2018] also experiments with a deviation-based technique which is the univariate Gaussian predictor, to find the parameters of the Gaussian model. It is trained on time series and uses *Maximum Likelihood Estimation* (MLE) to approximate the parameters of the Gaussian model, which are the mean and variance. The model classifies each new acquisition x_i based on the probability of that value in the distribution $p(x_i)$. Any new observation with a probability value less than a certain set threshold is then considered anomalous.

2.5.4 . Clustering-based methods

Clustering-based approaches mainly rely on clustering techniques to identify anomalous data. This category of techniques groups similar data and patterns in clusters. The fundamental principle behind clustering-based anomaly detection is that anomalies exhibit distinct characteristics that set them apart from normal data measurements.

The authors of [Giannoni et al., 2018] present their clustering-based approach: the *local density cluster-based outlier factor*. It is an extension of the *Cluster-Based Local Outlier Factor algorithm* (CBLOF). It works by clustering samples

using any clustering algorithm (e.g. k-means). It then separates large clusters from small clusters (hyper-parameters to be pre-set). An average distance from the centroid is then computed for each large cluster. These distances are used to approximate the densities of large clusters. If a sample is placed in a small cluster, its outlier score is the ratio between its distance to the closest large cluster centroid and the average distance of points in that large cluster from the centroid. Otherwise, if a sample is placed in a large cluster, its outlier score is simply the ratio between its distance from the cluster's centroid and the average distance of points in the cluster from the centroid. Data points with an outlier score above a certain threshold are considered anomalous.

The work of [Yoseph and Heikkilä, 2019] conducts comparative experiments on real point-of-sales (POS) database between two clustering-based methods: the $k\text{-}means$ (KM) and the $fuzzy\ C\text{-}means$ (FCM). The main objective of the k-means algorithm is to partition n objects into k clusters so that the inter-cluster similarity as the distance between observations is minimum and the intra-cluster similarity is maximum. In brief, the clusters are first initialized by picking random cluster centers for the k clusters. The measurements are then assigned to clusters depending on their distance to the nearest cluster. After all the measurements have been assigned to a cluster, the centers of these clusters are updated. Finally, the two aforementioned steps are repeated until the cluster assignments converge.

Using k-means, an outlier score is computed after each iteration for all the data measurements. This outlier score is the ratio of the distance of a data point to the centroid divided by the mean or median distance of all cluster members to the centroid of the cluster. Outliers are detected according to two different rules. The first one considers the element with the highest outlier score as anomalous. The second one identifies all data measurements with an outlier score above a certain threshold to be anomalous.

The other clustering algorithm studied by [Yoseph and Heikkilä, 2019] is the fuzzy C-means clustering algorithm introduced by [Bezdek et al., 1984], it allows a point to belong to more than one cluster in any of the $j = 1, .., k$ clusters. However, it associates a value $\mu_j = [0, 1]$ that determines the degree of its belonging to the cluster j. Membership of the data point i to k clusters are indicated with vector μ_{ij}. For a given data point i, the vector μ_{ij} contains for each cluster j a value of zero if the data point does not belong to cluster j, or a value of one if it belongs to cluster j. A data point i with several $\mu_{ij} > 0$ may indicate outliers in the spaces between cluster centers. A data point close to the cluster center has a high-degree of non exclusive membership in that cluster meaning that it most probably belongs also to other clusters, and generally has lower memberships in other clusters. As for a data point that is farther away from the cluster center, the non-exclusive membership in the cluster in question is lower but may also have memberships in other clusters.

The authors of [He et al., 2003] employ a clustering algorithm that focuses on finding clusters as much as assigning a cluster-based outlier score to each data measurement in the dataset. The authors use the Squeezer algorithm introduced by [He et al., 2002] for clustering the data. They propose definitions of their notions of small and large clusters. To detect outliers, the authors leverage the concept of "local outlier" proposed by [Breunig et al., 2000b] to introduce the notion of a *cluster-based local outlier factor* (CBLOF) that can determine the degree of deviation of a record. All data measurements in the dataset are associated with an outlier score. To assign an outlier score to a record t, the distance between the record and the closest large cluster is computed because large clusters are assumed in this work to have a smaller probability of being anomalous. If the record t already belongs to a large cluster, the outlier score is the distance to the centroid of its cluster. For any record t, the cluster-based local outlier factor of t is defined as follows.

$$CBLOF(t) = \begin{cases} |C_i|^* \text{ min(distance}(t, C_j)) \; where \; t \in C_i, C_i \in SC \; and \; C_j \in LC \text{ for } j = 1 \text{ to } b \\ |C_i|^* \text{ distance}(t, C_i) \; where \; t \in C_i, \; and \; C_i \in LC \end{cases}$$

Where C_i is a cluster, b is the threshold of large cluster, SC is a small cluster, and LC is a large cluster.

2.5.5 . Distance-based methods

These approaches rely on the distances between the data measurements to determine if a measurement is anomalous or not. Distance-based approaches were introduced to overcome the problems raised by the statistical approaches, such as the assumptions about the underlying data distribution.

The work of [Bakar et al., 2006] performs a comparative study of statistical-based techniques with the distance-based technique discussed in the following. Outlier detection is done using this technique based on two parameters: parameter (p) and distance (d). The authors first compute the distances d_1 of each data point from all the other existing data points. The maximum distance value recorded d_2 between any two data points is then identified. A threshold distance d_3 is determined based on the maximum distance value d_2, where the threshold is a value that is smaller than d_2. To determine the value of parameter p, d_1 and d_3 are compared, and if if d_1 is greater or equals d_3, then p is assigned the value of if d_1, otherwise, it is assigned the value of if d_3. Another threshold value t is later determined and compared with p in order to identify the outliers in the data. The Manhattan Distance is used to compute the distances in this approach as follows.

$$d(t_i, t_j) = \sum_{h=1}^{k} | \, (t_{ih} - t_{jh}) \, |$$

where $t_i =< t_{i1}, ..., t_{ik>}$ and $t_j =< t_{j1}, ..., t_{jk>}$ are tuples in a database.

With this distance-based approach, distances are compared to a simple data threshold determined based on the maximum distance value, which is able to

detect obvious anomalies but will not be able to detect pattern anomalies that are within the range value.

The work of [Tran et al., 2016] presents a comparative evaluation study of several distance-based outlier detection methods on data streams based on the definition of distance-based outliers presented in [Knorr and Ng, 1998]. The definition of [Knorr and Ng, 1998] stipulates that given a dataset D, a count threshold $k(k > 0)$ and a distance threshold $R(R > 0)$, a distance-based outlier in D is a data point that has less than k neighbors in D. The authors of [Tran et al., 2016] evaluate several distance-based approaches on synthesized and real datasets in similar conditions. The evaluation metrics are the CPU time and the peak memory requirement.

The data streams are considered to be received in batches and examined using a sliding window. The problem addressed in the studied approaches is: $what\ happens\ if\ one\ point\ has\ k\ or\ more\ neighbors\ in\ one\ slide\ but\ does\ not\ have\ k\ neighbors\ in\ the\ next\ slide$? These distance-based outlier detection for data streams (DODDS) approaches are described in the sequel.

The work of [Angiulli and Fassetti, 2007] introduced the Exact-Storm algorithm that stores for each data point o, up to k preceding neighbors of o in a preceding neighbors list (pn) denoted by $o.pn$, and list $o.sn$ stores the number of succeeding neighbors (sn) of o. This technique employs an index structure to store data points that supports range queries that find neighbors of data point o. A range query is a type of database query that retrieves data within a specified range of values. It is commonly used to extract subsets of data that fall within a particular range. When a slide of data points expires, data points are removed from the slide and moved to the preceding neighbors list (pn). After the range calculation in the new slide, outliers are verified by verifying if o has less than k neighbors, including succeeding neighbors and old non-expired preceding ones.

The work of [Yang et al., 2009] proposes the Abstract-C technique that uses the list $o.lt - cnt$ to store the neighbors of o in every sliding window point o is in. It also employs an index structure to store data points that support range query. To detect an outlier, the algorithm checks for every active data point o. If it has less than k neighbors in the current window, it is an outlier.

The work of [Kontaki et al., 2011] proposes the Micro-Cluster Based Algorithm (MCOD) where micro-clusters are formed instead of employing a range query. A micro-cluster has no less than $k + 1$ data points, of radius $R/2$, and is centered at one data point where R is a distance threshold. All points in micro-clusters are inliers. A new coming data point o, can either join another micro-cluster, form its own cluster (it being the center) if it can find at least k new points are found the PD. The PD is a list where data points not belonging to any micro-cluster, meaning they could either be outliers or inliers and have

neighbors from separate micro-clusters, are stored.

Finally, the last technique studied in the work of [Tran et al., 2016], Thresh-LEAP, is introduced in the work of [Cao et al., 2014]. Thresh-LEAP reduces the expense of the range query here by not storing data points in the same index structure and each slide having a separate, smaller index which reduces the expensive range query cost and facilitates the minimal probing principle. The intuition behind this technique is that it first looks for the succeeding neighbors and then subsequently the preceding neighbors per slide in a reverse chronological order. Data points keep the number of their neighbors from a slide in $o.evil$ and the number of succeeding ones in $o.ns$. Once a new data point o arrives, the technique adopts the minimal probing principle by finding o's neighbors in the same slide and the next slides until k neighbors are found. In worst-case scenarios, all the slides are probed. $o.evil[]$ and $o.ns$ are updated after probing, and o is added to the trigger list of each probed slide.

The approach presented in [Zheng et al., 2017] tackles the problem of spatial outlier detection with context. The authors developed a local model that leverages contextual neighbors of the data samples to predict the value of a current data sample using kNN kernel regression with Gaussian kernel weight. The difference between the predicted value and the ground-truth is then computed to represent the global outlier score assigned to the data sample. Then, local confidence is defined and combined with the global outlier score to provide the local confidence outlier score that is assigned to each data sample signifying its probability of being an outlier.
Each data point in the studied dataset has a contextual attribute vector $x_i \in \mathbb{R}^d$ (including spatial coordinates) and a behavioral attribute value $y_i \in \mathbb{R}$.

Adopting the first Law of geography by Waldo Tobler [Tobler, 1970] that says: "*everything is related to everything else, but near things are more related than distant things*," they first find the contextual neighbors for each data sample, and then use the behavioral attribute of these neighbors to predict the behavioral value for the current data sample in the study. The kNN kernel regression is used:

$$\hat{y_i} = \Sigma_{j \in N_i} w_i y_j$$

where w_{ij} is the Gaussian kernel weight:

$$w_{ij} = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{d_{ij}^2}{\sigma^2}\right)$$

where d_{ij} is the distance between two samples, σ is the standard deviation of the distance distribution. N_i denotes the set of k-nearest contextual neighbors of a sample z_i.

This approach then uses robust metric learning using the Mahalanobis distance between two data points.

2.5.6 . Neural network-based methods

Neural network-based anomaly detection approaches leverage the power of neural networks to detect anomalous patterns for time series. These approaches use different neural networks, such as autoencoders, recurrent neural networks (RNNs), and generative adversarial networks (GANs).

Autoencoder and GAN-based anomaly detection approach

The authors of [Audibert et al., 2020] developed an unsupervised anomaly detection approach based on adversely trained autoencoders. The autoencoder architecture makes it possible to learn in an unsupervised way on non anomalous data. The use of adversarial training and its architecture allows it to isolate anomalies while providing fast training.

The autoencoder [Rumelhart et al., 1986] is composed of an encoder E and a decoder D. The encoder part E takes an input X and encodes it into a set of latent variables Z. Then, the decoder part D, takes as input this set of latent variables Z and attempts to decode it back into the input as a reconstructed set R. The objective function of the autoencoder neural network is as follows Equation 2.9:

$$\mathcal{L}_{AE} = \|X - AE(X)\|_2 \tag{2.9}$$

Where AE(X) = D(Z), Z = E(X), and $\|.\|_2$ denotes the L2-norm. For anomaly detection, the trivial approach is for autoencoders to use the reconstruction error as an anomaly score and set a threshold δ where points having an anomaly score above the threshold are said to be anomalies [Audibert et al., 2020]. However, this approach is not efficient if an anomaly has a value not too far away from the non anomalous data points. Hence, [Audibert et al., 2020] proposes combining the autoencoders architecture with the adversarial training from the generative adversarial networks (GAN)s [Goodfellow et al., 2014], which trains the network to isolate the anomalies during the training phase. The architecture of the method proposed by the authors of [Audibert et al., 2020] is composed of three elements: an encoder network E and two decoder networks D_1 and D_2. When applied to time series, the series is transformed to windows of a specific size, then the training and detection are applied on windows instead of data points. The training is conducted over two phases. The first phase consists of both decoders D_1 and D_2 making an attempt to reconstruct the initial input with the objectives as follows:

$$\mathcal{L}_{AE_1} = \|X - AE_1(W)\|_2$$
$$\mathcal{L}_{AE_2} = \|X - AE_2(W)\|_2$$

Where each decoder attempts to reproduce the initial input window W.

The adversarial training is the second phase of the training during which the output of the first decoder AE_1 is encoded by the encoder E and then passed to AE_2. The objective of AE_2 here is to maximize the difference between the real

data and the data coming from AE_1 in order to distinguish which is real input data and which is not. The objective of AE_1, on the other hand, is to fool AE_2 by minimizing the difference between the real input W and the output of AE_2. The training objective function is as follows:

$$min_{AE_1} max_{AE_2} \|W - AE_2(AE_1(W))\|_2 \tag{2.10}$$

Where W is the input window W.
During the anomaly detection phase, an anomaly score is defined to each unseen window as:

$$\mathscr{A}(\widehat{W}) = \alpha\|\widehat{W} - AE_1(\widehat{W})\|_2 + \beta\|\widehat{W} - AE_2(AE_1(\widehat{W}))\|_2 \tag{2.11}$$

where $\alpha+\beta$= 1 and are used to parameterize the trade-off be- tween false positives and true positives.

LSTM-based anomaly detection approaches

The authors of [Braei and Wagner, 2020] surveyed several neural network-based approaches. One of the approaches surveyed is an approach based on *long short-term memory* (LSTM), first introduced by [Hochreiter and Schmidhuber, 1997]. LSTM belongs to the *recurrent neural network* (RNN) architectures with feedback connections that enable them to output information to the next input in the sequence. LSTMs were designed to avoid the long-term dependency problem of conventional RNNs. The output of a neuron of a recurrent neural network is the following:

$$y_t = \phi(x_t^T.w_x + y_{t-1}^T.w_y + b) \tag{2.12}$$

LSTMs, like all recurrent neural networks, have a chain-like repeating structure. Each cell in the LSTM network is shown in Figure 2.2. The authors of [Malhotra et al., 2015] stack two hidden layers of LSTM networks to predict the values of l upcoming timestamps of a univariate time series X_T = x_1, ..., x_T. The authors of [Chauhan and Vig, 2015] also propose an approach where the probability distribution of the prediction error is used to predict whether a data point is anomalous.

For both approaches, the error value e_i of each prediction x_i where i $\in 1, ..., T$ is computed. Then, the error vector of all predictions is used to fit to a normal distribution $\mathcal{N} = \mathcal{N}(\mu, \Sigma)$ using the *Maximum Likelihood Estimation* (MLE) to determine the values of μ and Σ. Finally, an anomaly threshold δ is determined during the training phase to distinguish anomalous data points from normal ones.

2.5.7 . Analysis

While basic statistical methods work very well in detecting anomalies for spikes and noise anomalies, they cannot be as effective with data with high variations. They can be too simple to detect hidden pattern anomalies, such as contextual anomalies that can only be detected while considering their context. For example, in the context of bank transactions, a customer making several

money withdrawals from an ATM may seem normal as the customer has already withdrawn that amount of money before, but when the consecutive transactions are considered together, the pattern that the customer is exhibiting is anomalous and could reveal bank card theft.

Deviation-based methods are methods that predict the value and compute its deviation from the observed one in order to detect an anomaly. This family of methods requires a clean training dataset that the model can learn non-anomalous data measurements. The discussed distance-based techniques perform well for real-time detection. However, they are unable to detect pattern anomalies. Neural network-based techniques are gaining remarkable attention in the literature recently for anomaly detection. However, neural network methods require a clean or a semi-clean dataset and are highly sensitive to their training data.

The techniques from all the different families rely on existing data either from the same sensor or from other nearby sensors to detect an anomaly. Data measurements collected by low-cost mobile sensors can sometimes have good quality and, at other times, have poor quality. This is due to the faulty nature of these nomadic sensors that makes them prone to quality issues that can affect their performance, such as manufacture defects, points of failure, battery drainage, etc. If anomalous data measurements are compared with data that is already of poor quality, the result of the detection will be poor because these measurements could be faulty in the first place and are not reliable. In order to improve the quality of the detection of anomalies, we could consider the quality of the sensors taking the measurements at the time they were taken. This allows for better detection because the measurements with poor qualities are given lower priorities than those with higher ones in order to limit their impact on the detection process.

As some anomalies can only be detected when considering the context they are in, it is sometimes impossible to detect an anomaly if we have no information about the context surrounding it. For example, in an air quality measuring context, assume we know from the existing data that the sensor holder is only exposed to a certain pattern of air pollutants when they take the metro on their way home from work. If this same pattern is recorded on the user's way back home from work and we do not have the context information that indicates that the user did not take the metro that day on the way home, the pattern will not be identified as anomalous, while the truth is that it is.

The detection of anomalies could be improved using the quality of the sensors and the contextual information. These information can be integrated in the similarity function to group similar data measurements together. The quality of the sensor can also be integrated in the anomaly detection process knowing that groups with higher quality measurements are less likely to be anomalous.

2.6 . Quality-based Data integration

In the preceding sections, we have introduced various dimensions of data quality, both for the mobile crowdsensing context and for other contexts. Our focus has been on two prevalent data quality challenges in mobile crowdsensing environments, namely missing values, and outliers. We have presented approaches from the literature that addressed these issues and proposed algorithms and tools for data quality improvement. The data gathered by the sensors will eventually be integrated for computing indicators and analysis. It would be interesting to include the quality of the data in the analysis process.

The main objective in sensor data integration systems is to compute indicators and extract insights from the integrated information coming from several sources. If we know what is the quality of each measurement coming from the sensor, the question of how we can consider this quality during the integration of the data and during the computation of the indicators arises. The intuition is that considering the data quality should lead to better indicators. This problem of integrating the quality of the data in the integration of data coming from multiple data sources has been the focus of some works in the literature.

Some systems have studied the way quality can be integrated in data analytics. An example of such systems is the *Spatial On-Line Analytical Processing* (SOLAP) systems. Spatial OLAP systems [Bédard et al., 2006] can be defined as software that allows rapid and easy navigation within spatial databases and offers many levels of information granularity. The SOLAP (Spatial OLAP) concepts support the multidimensional paradigm and enriched data exploration based on an explicit spatial reference represented on maps. SOLAP systems can provide powerful analytical capabilities for exploring and visualizing spatial and temporal aspects of mobile crowdsensing data, enabling users to gain deeper insights and make informed decisions based on the geospatial context and evolving patterns within the data.

In this section, we focus on two problems. The first problem is how data quality is represented. For this, we will present approaches that propose quality models for data warehouses. The second problem is how quality is taken into account for data aggregation.

2.6.1 . Quality Models for Data Warehouses

With the rise of new geographical data acquisition technologies, the number of available temporally geo-tagged datasets is also rising. The urge for proper tools for storing and representing this data is also increasing. The authors of [Berrahou et al., 2015] propose a multidimensional model star schema of their data while including quality in the fact table. The authors present their multidimensional data model showing the fact table that measures hydro-ecological watercourse sampling data. The authors integrate thematic, temporal, and spatial accuracy in the data modeling process, where each facet of the accuracy is

attributed to one or more different data packages. The quality weight is stored as a measure directly in the Fact table Quality is later weighted by the facts in the aggregation-based analysis, which will be discussed in further detail in subsection 2.6.2.

The work of [Jarke, 2003] defines data quality in data warehouses at several levels; the authors define the quality at the design and administration where quality schema with all the quality dimensions is proposed. Another level is the software implementation quality level, where the authors adopt standards proposed by ISO 9126. Finally, there is the quality at the data usage level, where a schema for data usage quality dimensions is proposed and explained. Moreover, the authors define a quality meta-model for data warehouse quality, where quality goals are linked to a set of quality queries used to decide whether or not a quality goal is attained.

The authors of [Devillers et al., 2005] first propose a model named *Quality Information Management Model* (QIMM) that allows the management of spatial data quality information within a datacube. The spatial data quality information stored within the QIMM model are later manipulated using SOLAP to allow the expert to navigate into quality dimensions and to intersect them for any level of detail. The authors integrated the quality values into the data models. They proposed approaches that allow drilling down and rolling up on different granularities of data quality dimensions along with their corresponding data. For example, drilling down from a general data quality value to a specific quality value of logical consistency or semantic accuracy.

The authors of [Devillers et al., 2007b] developed a prototype to support data experts in the assessment of the fitness of certain data for an intended use based on the QIMM data structure within a spatial data context. They store the quality information within the multidimensional database. The authors develop a tool that show some quality indicators on the data on several granularity levels of the data. This approach allows users also to explore data quality along a quality indicator hierarchy. The quality indicators in the dashboard can be drilled down and rolled up. It allows users to explore the quality of the data as well as the aggregated data. For example, if the user can see quality information at highl level such as the completeness of the data, the user will be able to drill down into factors of data completeness, such as spatial completeness.

The authors do not provide one single way to aggregate quality information but rather provide an approach that helps data users select the aggregation process that best fits their needs.

2.6.2 . Quality aggregation operators

The authors of [Boulil et al., 2013] propose a unique framework to represent, at the conceptual abstraction level, integrity constraints in spatial warehoused data context where the integrity constraints are defined on the data, aggregation, and

spatial-multidimensional query levels. The authors defined three integrity constraint (IC) classes: the *data IC*, the *aggregation IC*, and the *queries IC*. These integrity rules were specific to the context of meteorological spatio-temporal data and were implemented using object constraint language (OCL). The data integrity constraints ensure the logical consistency and completeness of the spatial data. The data IC can be defined on all elements of the spatial data warehouse (SDW), such as facts and members. The aggregation integrity constraints guarantee correct and meaningful aggregations of the measures. There are two defined sub-types of aggregation integrity constraints; the first one is semantic constraints that check if an aggregation function applies to the measures according to the semantic nature and the type of the measures, the aggregate functions, and the dimensions. The other sub-type defined is schema constraints that impose conditions that must be satisfied by dimension hierarchies and respect the relationships between the dimensions and the facts in order to avoid redundancy and incompleteness of the aggregations. And finally, the authors also defined query integrity constraints that refer to conditions that ensure that SOLAP queries are valid to avoid any misinterpretation resulting from empty query results.

The work of [Berrahou et al., 2015] is in the environmental context. The authors work on integrating quality in the aggregation-based querying on the spatial on-line analytical processing information system. The thematic, temporal, and topological accuracy of each record in the system is computed, then the computed quality value is stored in the fact table. All records in the fact table can be queried on several dimensions. Each value is weighted by the three quality values computed. The quality values are computed by function weighting and combining several logical rule-based constraints to validate the consistency and completeness of the record. Temporal accuracy is a constraint that checks whether the date and time attributes are not null and are valid. The thematic dimension of the accuracy data quality factor is computed compared to a set of domain-specific constraints and characteristics regarding the consistency of multiple physio-chemical parameters that the record/measurement needs to fit. The topological/spatial accuracy is computed as the distance between a measurement and a reference water site. Finally, the authors have integrated these data quality values in the aggregation queries on the data. This approach used different data quality factors as filters while executing queries on the data.

The authors of [Bimonte et al., 2006] propose an extension to a multidimensional data model that can support complex objects as measures to integrate spatial data. The approach also focuses on aggregation measures and designing ad-hoc aggregation functions. The defined ad-hoc aggregation approach supports semantics for the correctness of the aggregation and dependencies across aggregation functions. The authors define entity, hierarchy, and cube schemas in the geographical spatial context. Also, the authors focus on defining the notion of aggregation that allows the creation of ad-hoc aggregation functions that support hierarchies.

To our knowledge, no works integrate data quality when aggregating sensor data in mobile crowdsensing environments. The data measurements coming from low-cost nomadic sensors can sometimes have very poor qualities and high quality at other times. Therefore, assigning equal weights to all the available measurements will compute aggregations that are also of poor quality.

2.7 . Conclusion

In this chapter, we studied approaches defining different quality dimensions in general contexts as well as in the context of mobile crowdsensing environments with sensor data. The existing definitions of the data completeness quality dimension are insufficient and lack a clear representation of the various aspects of data that characterize the context. Also, the existing metrics evaluate the data completeness in a quantitative manner only rather than qualitative. These existing evaluation metrics require improvement to better assess the completeness aspects. In order to improve the data completeness, many works proposed several approaches to generate the missing values using data imputation techniques [Troyanskaya et al., 2001], [Li et al., 2009], [Yi et al., 2016]. The generation of missing values that are accurate and can replace big chunks of missing data are still research problems to be studied and improved. The existing approaches are completely oblivious to the quality of the sensors taking the data measurements, which makes the imputation also quality-oblivious.

Another main flaw of the sensor data is the presence of undetected anomalies. There are numerous works targeting anomaly detection for sensor data, but obviously, the state-of-the-art still lacks a clear guide to which family of techniques is most suitable to which context and what data. Detecting anomalies is still a topic of research with the rise of data acquisition sources that may not be completely reliable. Even though in the past recent years, there have been more works trying to leverage the context to help improve predicting or detecting an anomaly in the data, the contextual data are still vaguely defined and incomplete.

During the acquisition process, big chunks of the data could be lost, noise could be introduced to the data, there could be calibration issues with the measuring sensor, etc. Analytics and insights that are oblivious to the quality of the sensors taking the data result in indicators that are aggregation or queries that do not know the quality of the computed aggregates.

In this thesis, we target some of the aforementioned discussed problems. We introduce quality factors adequate for mobile crowdsensing environments and provide the associated metrics. We characterize different factors of completeness and propose metrics to evaluate each of them in chapter 3. We also tackle in chapter 3 the data completeness improvement by extending existing data imputation techniques with the quality of the sensor to impute a single missing value measurement at a time.

In chapter 4, we address the accuracy data quality dimension. We propose an anomaly detection approach within mobile crowdsensing environments considering data about the quality of the measuring sensors and some other contextual information. In our approach, we propose an anomaly detection approach that takes data quality into account. We also take into account the contextual information if available.

And finally, we address in chapter 5 some ways of integrating data quality in sensor data analytics. We address the problems related to taking data quality into account during the analysis of sensor data. To this end, we introduce quality-based data aggregation operators. We also characterize the quality of a computed aggregate given the quality of the underlying data measurements.

3 - Data Completeness in MCS

3.1 . Introduction

Data completeness is a data quality factor that measures if a dataset is not missing information. It is applicable in a wide variety of application contexts. Data completeness is defined in relational data as the extent to which the table represents the corresponding real-world objects[Batini and Scannapieco, 2006]. Studying the impact of data completeness, assessing, and improving it are ongoing research topics in different application contexts [Azimi and Pahl, 2021], [Liu et al., 2017]. For example, the authors of [Azimi and Pahl, 2021] study the impact of data completeness levels on machine learning models, and the authors of [Liu et al., 2017] survey the literature for existing problems of data completeness in the healthcare sector. To the best of our knowledge, there are no works that study the relevant quality dimensions in mobile crowdsensing environments for data completeness. We study in this chapter the characterization, assessment, and improvement of the data completeness quality factor in mobile crowdsensing environments. Mobile crowdsensing (MCS) is an emerging sensing model which primarily depends on the strength of the people's sensor-enabled mobile devices to collect the data for a particular acquisition task [Ma et al., 2014, Brahem et al., 2021]. Data collected by sensors in this context are spatiotemporal series that are time series where each data measurement is tagged by coordinates of the geographic location it was taken at.

Several metrics have been proposed to evaluate data completeness for different data types. For example, the data completeness quality dimension is evaluated in relational databases as the percentage of null values if the data contains null values. Otherwise, data completeness is evaluated compared to a reference dataset and has missing records [Batini and Scannapieco, 2006]. Either way, these definitions are insufficient to evaluate data completeness for sensor data in a mobile crowdsensing environment because, in the case of the presence of null values, it is possible to have a table that does not contain any null value and still not be complete because there are records that are missing. Otherwise, comparing the actual data to a fixed reference is also insufficient because they do not capture the several understandings of data completeness that exist in mobile crowdsensing environments. Hence, this existing definition is also insufficient.

Missing data is a very commonly encountered problem when dealing with data coming from nomadic sensors. The missing values or chunks of the data degrade the completeness of the data if left untreated. Analytical studies based on data with untreated missing values can be misleading and unreliable. Hence,

existing works employ data imputation techniques as one way to generate the missing values [Khayati et al., 2020]. Data imputation techniques are techniques that can generate and fill in a missing measurement value based on the other adjacent existing measurements. For example, a data imputation technique T may generate a missing measurement in a time series at timestamp t by looking at the other existing measurements at timestamps t-1 and t+1. Sometimes an imputation technique T considers spatially adjacent existing measurements that are geographically close to the missing one. The problem with existing approaches is that they neglect the faulty nature of the sensors which may result in poor quality data. This makes sensors that are having quality issues at the moment be treated equally to the perfectly working ones. Measurements from sensors with bad quality may tend to have values that are further away from the ground truth than those from good-quality sensors.

This chapter characterizes the data completeness quality dimension in a mobile crowdsensing environment (MCS) with its different factors. We introduce a multidimensional data model to characterize data captured within a mobile crowdsensing environment that helps us visualize the data analytically. We define three data completeness factors: *sensor completeness*, *temporal completeness*, and *spatial completeness*. We propose a set of metrics for the evaluation of these different factors. We then present some solutions for data completeness improvement. We define the quality of the sensor in the context and present some possible metrics that can assess the quality of data measurements coming from these sensors. We extend three existing data imputation approaches from different families, ST-MVL, SVDImpute, and KNNImpute, to improve the completeness of the data. The quality-aware data imputation approach aims at considering the quality of the measurements during the imputation by assigning a lower priority to low-quality measurements and a higher priority to high-quality measurements to improve the quality of the imputation.

In the remainder of this chapter, we provide a motivating example for data completeness in mobile crowdsensing environments in section 3.2. Then, the multidimensional model is presented and defined in section 3.3. The three factors of data completeness are defined in section 3.4 where we propose the metrics for their evaluation. The quality-aware data imputation approach is presented in section 3.5 where we introduce and define sensor quality as well as our proposed extensions of the three aforementioned techniques. Finally, we evaluate the proposed metrics as well as our quality-aware extended approach in the experiments discussed in section 3.7, and conclude the chapter in section 3.8.

3.2 . Motivating example

According to the authors of [Batini and Scannapieco, 2006], data completeness has been defined as "the extent to which data are of sufficient breadth, depth, and scope for the task at hand". The authors propose several metrics to evaluate data completeness in relational databases. The first one is the presence of null values in a given table or column. The second metric is the comparison of the tuples present in the database to an existing set of reference tuples [Batini and Scannapieco, 2006]. The first metric is insufficient because it is possible to have a table that does not contain any null value and is still incomplete if, for example, an entire record is missing. The second metric is also insufficient because it does not consider the different aspects of data taken in a mobile crowdsensing context. For example, the measurements could happen to be all taken according to the reference dataset but not according to another aspect of the data such as the spatial aspect. This is demonstrated in the following example 3.2.1.

Example 3.2.1. Assume we have data collected by nomadic sensors measuring a particular physical phenomenon with time and geographic location. The collected data measurements by a sensor are represented by a series $X_n = \{v_1, \ldots, v_n\}$. We assume that the measurements are taken at a frequency f for the considered time period. If all the sensors have provided the measurements according to f, we could say that the data is complete. However, if the measurements happen to all be taken in a very small partition of the studied area, this means that the measurements do not cover the considered area, and are hence incomplete with respect to the spatial dimension.

Consider, for example, the table in Figure 3.1 which shows a sample of the measurements from one sensor. This table contains the timestamp at which the measurement was taken, the value of the measured element, and the longitude and latitude indicating the location of the sensor at that time. Let us consider that data completeness is evaluated as the proportion of Null values in the table. We can see from Figure 3.1 that there are no such values for any of the records in the table, and we can therefore say that our data is complete.

timestamp	value_from_sensor	latitude	longitude
2019-06-16 22:34:00+00	9	48.8561134338	2.37139344215
2019-06-16 22:35:00+00	9	48.8560180664	2.3714621067
2019-06-16 22:36:00+00	10	48.8560180685	2.37128782272
2019-06-16 22:37:00+00	12	48.8561134338	2.37139344215
2019-06-16 22:38:00+00	13	48.874317	2.302189
2019-06-16 22:39:00+00	13	48.87352	2.303082
2019-06-16 22:40:00+00	8	48.859813	2.299612
2019-06-16 22:41:00+00	10	48.859231	2.300347
2019-06-16 22:42:00+00	13	48.858699	2.300915
2019-06-16 22:43:00+00	11	48.8588929716686	2.30260951056391

Figure 3.1: Snapshot of the data captured by sensors.

After plotting these data measurements on a map as shown in Figure 3.2, we can see that these measurements cover only three cells in the studied area. We

also notice that no measurements are recorded in the remaining cells of the grid. This means that even though the data has no null values, the data measurements however are not covering the spatial area being studied which implies that the data is not complete spatially. Therefore, the traditional metric of computing the proportion of null values of the data is not sufficient for our context.

We consider another example where we assume that the rate of measurement of the sensor is 1 measurement/second. This means that we expect 10 minutes $\times$ 60 measurements/minute $=$ 600 measurements during this 10 minutes period of our study. Even though the data has no null values in Figure 3.1, there are 590 missing measurements in that table. Hence, the data in our table is incomplete.

The examples presented above show that the existing completeness definitions and metrics are not appropriate to capture all the dimensions of data completeness in MCS environments. In the following section, we will present a multi-dimensional model for storing pollution measurement data generated within a mobile crowdsensing (MCS) context, and we will discuss the different factors of completeness in this context.

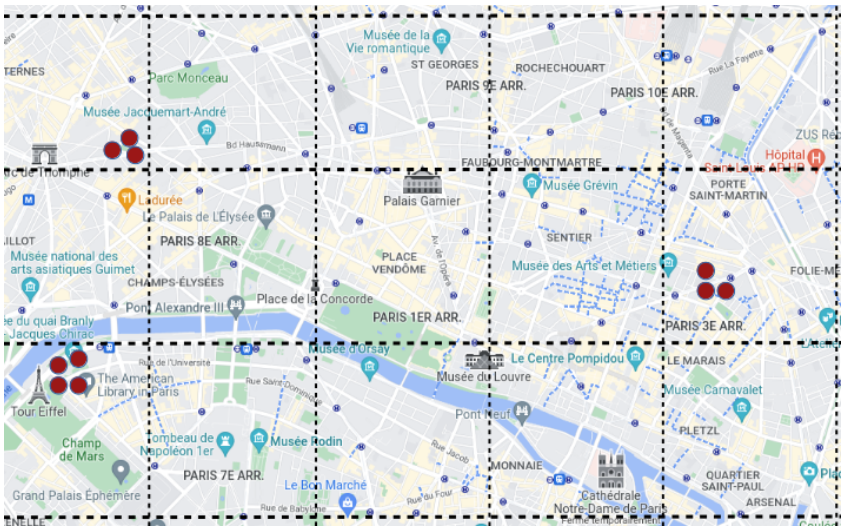

Figure 3.2: Map showing the spread of the pollution measurements over the grids of a given area.

3.3 . Multidimensional data model in MCS environments

In order to understand the characteristics of data in a mobile crowdsensing environment, we propose our multidimensional data model presented in Figure 3.3. This model helps us understand the different dimensions of the data, it simplifies its analysis and helps visualize the different quality factors concerning this data. The multidimensional data model presents the measurement (fact) and the dimensions of sensor data. The measurement value in fact table represents a physical

element measuring a real-life phenomenon captured within the context of a mobile crowdsensing environment (MCS). We will use this model to illustrate the different factors of data completeness.

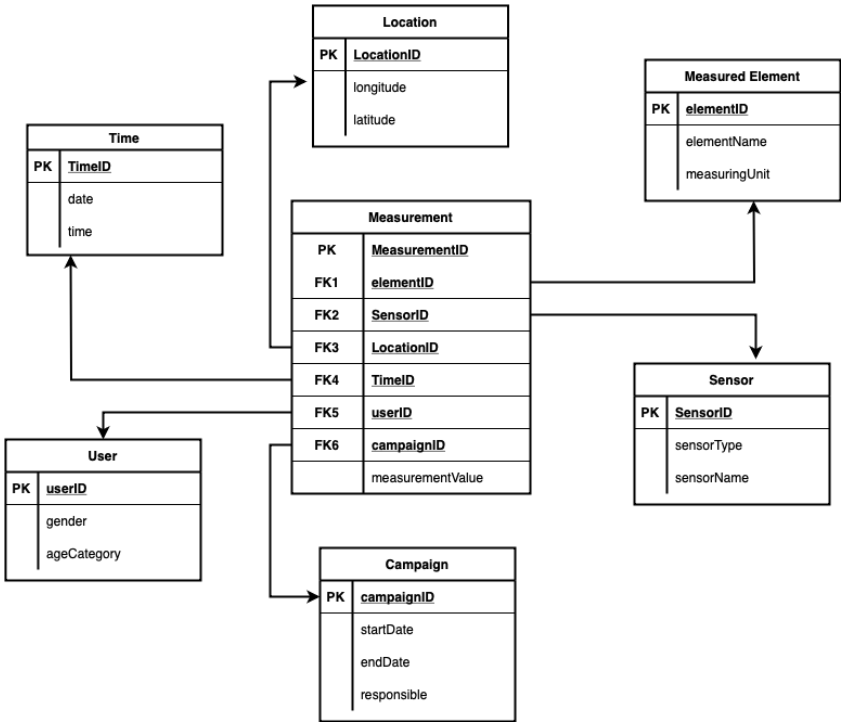

Figure 3.3: A Multi-Dimensional Model for Pollution Data.

3.3.1 . Sensor data in a MCS environment

A sensor s_i in a mobile crowdsensing environment collects a set of measurements $X_n = \{v_{i1}, v_{i2}, \ldots v_{in}\}$, where v_{ij} is a measurement vector taken by sensor s_i at timestamp t_j, and measuring any physical real-world phenomenon. Each vector v_{ij} contains a measurement v and a set of r features that represent the dimensions of this measurement. The vector $v_{ij} = (v, f_1, \ldots, f_r) \in \{D_v \times D_1 \times \ldots \times D_r\}$, $r \in \mathbb{N}^+$ being the number of features each measurement has, D_v is the domain of the measurement value, and D_i $i \in 1, ..., r$, is the domain of a feature f_i in the vector v_{ij}.

Assume a context where a data measurement represents some physical phenomenon and the dataset comprises a set R of time series, $R = \{X_1, X_2, ..., X_n\}$, acquired by a set of sensors S = $\{s_1, s_2, ..., s_n\}$. Each sensor collects measurement values of any physical phenomenon alongside the timestamp and other features of the data. Suppose that the measured physical element is a given air pollutant. Each measurement vector v_{ij} is collected by a sensor s_i and has the following features.

The measurement value is the level of the measured phenomenon (e.g. meteorology, air pollution, water composition, online transaction, etc.). The measurement is taken at timestamp t_j which indicates the time at which the measurement was taken. The pair (lat,lon) indicates the spatial geographic coordinates where the measurement was taken. The activity a represents the environment where the measurement was taken. Finally, s_i represents the sensor unit that took the measurement.
For analysis purposes, the measurement vector v_{ij} has to have at least the following features: $v_{ij} = (v, s_i, t_j, (lat, lon), s_i)$ which is a general set of some features that represent a measurement in a mobile crowdsensing environment (MCS). More features could be added to model the data when characterizing the model to a specific application domain in the crowdsensing context.

3.3.2 . The data model

This subsection presents the multidimensional data model that describes data in a mobile crowdsensing environment. The model covers several dimensions that are typically found in sensor data, such as user, campaign, sensor, time, location and the physical phenomenon being measured.

Figure 3.3 depicts our multi-dimensional model. A single sensor reading is represented in the fact table Measurement. The value of the measurement is the attribute measurementValue which represents the value of the reading captured by the sensor of a certain measured element. The Measurement fact table has 6 dimensions that are defined as follows:

- *Campaign:* represents the duration of time, with a start and end date where a number of users carried sets of sensors to collect data.
- *User:* identifies the participant who was carrying the sensor that took this measurement; user-identity information is not saved for privacy reasons; the gender and age are recorded for analysis purposes.
- *Sensor:* represents the sensor that took the measurement, described by a sensor id, a type, and a name.
- *Location:* represents the spatial coordinates where the measurement value was taken.
- *Time:* gives information about the date and time when the measurement value was taken.
- *Measured Element:* provides information about the name of the physical phenomenon that is being measured.

To instantiate the data model with the features described in this model, a data measurement can be described by the vector $v_{ij} = (v, s_i, t_j, (lat, long), u, c)$. The measurement value of the vector v_{ij} is denoted by v, the sensor that captured

the measurement is denoted by s_i, the user that carried the sensor is represented by u, the timestamp at which the measurement was taken is denoted by t_j, the geographic location where the measurement was taken is represented by the pair (lat, long), the campaign c during which the measurement was taken.

3.3.3 . Quality factors for data completeness in MCS

Considering the model presented in Figure 3.3, we leverage the different dimensions of the data measurement to propose the data completeness factors of data in this context. Completeness in mobile crowdsensing environments has different facets, and there are several understandings of how completeness can be perceived and represented. The multidimensional model in figure Figure 3.3 helps us analyze the different facets and perspectives of completeness, we present five of them in the following.

Completeness of a campaign

The completeness of a campaign expresses the overall completeness of a campaign. It represents the extent to which the measurements expected during this campaign coming from all the sensors that were used by the participants are actually stored and available.

Completeness for a user

Completeness for one user over a period of time expresses the completeness of the measurements from all sensors carried by this user during their participation over a specific period of time. User completeness measures the extent to which the data expected from a user is complete considering the number of sensors the user carried and the duration over which the user carried the sensor.

Example 3.3.1. Consider two users user1 and user2 that carried each a sensor unit of the same type measuring the same physical element, for the same duration. User completeness of user1 and user2 are supposed to be equal given they have identical conditions. This might not always be the case in real-life scenarios because the usages of each one of the sensors by each user are different. If the user completeness of user1 is higher than that of user2, this could imply that maybe user2 turned off their sensor at some point. Hence, it allows us to compare the usage of the sensors by each user.

Completeness of a sensor

Sensor Completeness reflects the completeness of a single sensor unit over a specific preset period of time. During a certain time period, a sensor could be used several times by different users for variable durations. The study of sensor completeness

over a time period shows the extent to which this sensor has provided the expected measurements at the frequency it was supposed to deliver.

Example 3.3.2. Assume a sensor unit s_1 has been employed to collect data over a certain period of time T. If sensor s_1 is deactivated or turned off for the second half of the time within period T, then the sensor will only collect measurements for the first half of period T.

Completeness of a spatial area

Completeness for a spatial area represents the spatial coverage of a preset area. It indicates the spatial dispersion of the measurements over this area. The goal is to understand the way measurements are distributed in the considered area of study. It is also used to study how well the measurements cover the designated area.

Example 3.3.3. Assume we have a preset area A and a total of 100 measurements in this area. Suppose the reference completeness of any area measures the distribution of the available measurements equally over the different sub-parts of the designated area. If the 100 measurements are clustered in two small parts only of the area of study, the spatial completeness, in this case, would be very low.

Completeness of a time period

Temporal Completeness characterizes the extent to which a given period of time is covered by the collected measurements. This completeness factor measures whether the collected measurements were distributed across the different chunks in the time period or not. Temporal completeness depends on several elements such as the number of operating sensors, the physical element studied and its optimal reference, etc.

Example 3.3.4. Suppose we want to study the completeness of a time period P, and assume 10 sensors are expected to be capturing measurements during this period. If only 2 out of the 10 sensors were turned on during the first third and all 10 sensors were turned on during the last two-thirds of this time period, then the temporal completeness of the remaining two-thirds is higher than that of the first third.

3.4 . Completeness factors and their evaluation metrics

The data completeness quality dimension, like several other quality dimensions, is defined for various contexts and application domains. For example, the authors of [Batini and Scannapieco, 2006] defined data completeness for relational databases as "the extent to which the table represents the corresponding real world". However, this definition of data completeness is not sufficient for mobile crowdsensing environments because the data has several dimensions and the data completeness

could be considered from multiple points of view. This raises the need for a new definition that is more appropriate for the MCS context.

In this section, we define three data completeness factors considering the model described in Figure 3.3, sensor, temporal, and spatial completeness. These facets of data completeness reveal important knowledge about the different characteristics of mobile crowdsensing environments. The completeness factors provide a different perspective of completeness for every dimension of the data compared to existing definitions for traditional databases which are not multidimensional. We present our definitions of each of these factors, sensor completeness, spatial completeness, and temporal completeness consecutively. We give definitions, descriptions, and examples. We also present some quality metrics allowing to assess each completeness factor.

3.4.1 . Sensor completeness

Sensor completeness is a factor that expresses the completeness of the data captured and sent by a specific sensor unit s_i during a preset time period. Given the erroneous nature of some sensors, studying sensor completeness can show how reliable a sensor unit is over a certain time period T by giving information about the completeness of the data captured and sent by the sensor. Sensor completeness is a quality factor that captures the extent to which the measurements of a given sensor are complete over a certain time period.

A sensor unit can be used several times by different users and over one or more time intervals. A single sensor usage is carried out and associated with one single user over a certain duration. To study the completeness of one sensor over a certain period of time T, we have to study its completeness over the cumulative time duration within T regardless of the participant carrying the sensor. Hence, if a sensor has been used four times during a period T, we have to study its completeness over the accumulated time intervals within T where it was being used.

We evaluate the completeness of a specific sensor s_i as follows:

- Identifying all the usages of sensor s_i in the specified time period, as a sensor might have been used several times during the time period.
- Evaluating sensor completeness for each usage of s_i separately.
- Aggregating the computed evaluations of each usage to calculate the completeness of sensor s_i.

Definition 3.4.1 (Sensor completeness)**.** Sensor completeness represents the extent to which the measurements of a given sensor are complete over a certain period of time. The completeness for a single sensor s_i over a time period is evaluated as the number of available measurements over the required number of

measurements.

$$SenC_{s_i} = \frac{AM_{Si}}{RM_{s_i}} \tag{3.1}$$

Where AM_{s_i} is the actual number of measurements sensor s_i has taken during all its usages in the specified period of time, and RM_{s_i} is the expected number of measurements from sensor s_i during its usages in this time period.

The required number of measurements RM_{s_i} for a sensor s_i throughout a specific time period is defined as:

$$RM_{s_i} = \Sigma_{j=1}^{K} n_{s_{ij}} \tag{3.2}$$

Where K is the number of usages of sensor s_i over the specified time period, and $n_{s_{ij}}$ is the number of required measurements for sensor s_i in usage j.

For every single usage denoted j including the sensor s_i, $n_{s_{ij}}$ is computed as follows:

$$n_{s_{ij}} = f_{s_i} * D_{C_j} \tag{3.3}$$

Where f_{s_i} is the sampling rate of the sensor s_i and D_{C_j} is the duration of the usage j of the sensor.
To illustrate this evaluation of sensor completeness, let us consider the following example.

Example 3.4.1. Given a sensor unit s_1 measuring black carbon (BC) used by three participants within a time duration d, the sensor completeness of sensor s_1 over d is the ratio of the actual number of measurements observed by s_1, to the required number. The required number of measurements is defined by the sampling rate of the sensor. If we assume the required number of each usage of the sensor is 500 measurements, and the actual observed numbers are 350,200,480 respectively, then the sensor completeness of s_1 over d is (350+200+480)/500*3 = 0.69.

3.4.2 . Spatial completeness

Spatial completeness is a factor of completeness that describes the distribution of the measurements in the considered spatial area. It indicates how sufficient and comprehensive the current measurements are for a particular area. The optimal case can vary according to the requirements of the data in the application context. For example, an application could require for the measurements to be uniformly distributed over space. In this case, it is similar to the concept of data skewness [Belussi et al., 2018]. Another application could require for the measurements to be more concentrated in some areas rather than others because there is a higher population which means a better understanding of the studied element around highly populated areas.

Spatial completeness is the extent to which data sufficiently represents a specific spatial area, and it characterizes the coverage of this area considering

the available measurements. The spatial completeness of measurements does not necessarily mean the more the better. It only means that the available measurements are distributed over the area and they cover all of the parts of the studied area. Spatial completeness is quantifying the distance between the reference distribution and the actual placement of the measurements. We give an example of spatial completeness in the following.

The specification of a reference spatial coverage in order to assess spatial completeness can be based on different assumptions. We propose several interpretations for the required number of measurements for a specific area. This reference can differ from one application to another depending on the requirements. For instance, one interpretation that we propose in this chapter would be to divide the designated area into equal grid cells and compare the actual data to a uniform distribution of the measurements over the grid cells. This means that the measurements taken by the sensor are not grouped in a few portions of the area but are rather evenly distributed all over it. The spatial completeness would be comparing the actual distribution of the measurement with the reference distribution. For this particular interpretation, the evaluation steps of spatial completeness over the designated area are as follows.

- Dividing the area of study into equal-sized grid cells
- Computing the required number of measurements RM_{C_i} for each grid cell and evaluating the spatial completeness of each grid cell
- Aggregating the computed evaluations of each grid cell to compute the spatial completeness of the area of study

Different assumptions could be made in order to estimate RM_{C_i}, the required number of measurements in a given cell. Two of them are presented hereafter:

- **Assumption 1:** We consider as a reference, a uniform distribution of the measurements over the area of study A. This means that the number of measurements should be evenly distributed over the cells in the grid. Hence the required number of measurements would be:

$$RM_{C_i} = \frac{AM}{|A|} \tag{3.4}$$

  Where AM is the actual number of available measurements for the whole grid, and $|A|$ is the number of grid cells in area A.

- **Assumption 2:** We consider as a reference, a distribution of the measurements that takes into account the variation of measured element levels in the different cells of the area of study A. The variability of the measured element here will be studied from existing data (trends, seasonality, etc).

If for a given cell the data shows that there is a low variation of the measured element levels in all the spatial area represented by this cell, then the number of required measurements for this cell can be low without a loss of coverage. Conversely, if there is high variability in a given cell, then the required number of measurements should be higher to better represent this cell.

Once we have divided the area of study into grid cells, we compute the spatial completeness of each cell C_i as follows.

Spatial Completeness of a Cell C_i

After dividing the designated area of study into equal-sized grid cells, we compute the spatial completeness for each cell in the grid.

Definition 3.4.2 (Spatial completeness of a cell)**.** Spatial completeness of a grid cell C_i, denoted SC_i, is computed as follows:

$$SC_i = \frac{AM_{C_i}}{RM_{C_i}} \tag{3.5}$$

Where AM_{C_i} is the actual number of measurements in a grid cell C_i, and RM_{C_i} is the required number of measurements in a grid cell C_i.

The value of the spatial completeness of a cell SC_i ranges from 0 to 1. A value of 1 means that the available measurements are equally distributed over the different parts of the considered area. A low value represents the fact that the measurements are unevenly distributed over the area. It is worth noting that a high spatial completeness value does not represent the fact that a high number of measurements is available but that the available measurements, regardless of the quantity, are better distributed.

Spatial Completeness of Area A

After computing spatial completeness for each cell in the grid separately, the overall spatial completeness for the whole area of study A is computed by aggregating the spatial completeness of all the cells. This could be done in different ways, for example, using the average, the median, the minimum, or the maximum functions.

We propose two quality metrics to compute the overall spatial completeness.

Definition 3.4.3 (Completeness of an area using cumulative average)**.** Spatial completeness is the extent to which the available measurements cover a spatial area A. Evaluating the spatial completeness of an area A using the cumulative average computes the average of all cells' spatial completeness, as shown in the formula below.

$$SC(A) = \frac{\sum_{i=1}^{|A|} SC_{Ci}}{|A|} \tag{3.6}$$

Where SC_{Ci} is the spatial completeness of one grid cell C_i, and $|A|$ is the number of cells in the grid covering area A.

Definition 3.4.4 (Completeness of an area using completeness above threshold)**.** Evaluating the spatial completeness of an area A using the completeness above a threshold computes the proportion of cells having their spatial completeness above a given threshold *t*, as shown in the formula below:

$$SC_{(A)} = \frac{\sum_{i=1}^{|A|} \alpha_i}{|A|} \tag{3.7}$$

$$\text{where } \begin{cases} \alpha_i = 1 & if SC_{Ci} \geq t \\ \alpha_i = 0 & if SC_{Ci} < t \end{cases}$$

The difference between these two definitions is that the completeness of an area using the cumulative average considers the value of the spatial completeness of all cells in the area while computing the average while the completeness of an area using the completeness above threshold considers only the cells with a completeness value above the predefined threshold.

The following is an example of the spatial completeness of area A using the cumulative average to compute the spatial completeness of a cell.

Example 3.4.2. Given a specified area of study unit A with data coming from 5 sensors within a time duration d, the spatial completeness of area A over d is the ratio of the actual number of measurements observed by all sensors within the designated area A, to the reference number. If we assume that the reference assumption is a uniform distribution of the measurements, the actual number of collected measurements by these 5 sensors is 2000, the area is divided into 10 cells, and the actual observed measurements in each cell are 200, 100, 155, 5, 590, 50, 600, 300, 0, 0. The required number of measurements in each cell should be 2000/10 = 200, and the spatial completeness over area A is 0.45.

3.4.3 . Temporal completeness

Temporal completeness is a factor of completeness that describes the distribution of the measurements temporally compared to a reference. It indicates how sufficient and comprehensive the current measurements are for a certain period of time T. The reference can vary according to the requirements of the data in the application context. Temporal completeness is defined in our context as follows. Temporal completeness is another factor of data completeness that expresses the extent to which a considered period of time is covered by the available measurements.

On the one hand, sensors capturing measurements at a very high frequency may, at some point, add redundancy to the data, but on the other hand, a very

low frequency will lead to a number of measurements that is not sufficient. Therefore, we need a clear characterization of temporal completeness. High temporal completeness indicates a high coverage of the acquired measurements over a time period P. To assess temporal completeness, we divide a period P into n equal chunks and then compare the number of acquired measurements during time period P to a reference number of measurements defined for each chunk.
The evaluation of temporal completeness for a specified period of study is done as follows.

- Dividing the period of study P into equal-sized chunks of time D_i where i $\in \{i = 0, 1, 2..., n\}$ as it is shown in Equation 3.5.3
- Computing the required number of measurements denoted by RM_{Di}, and evaluating the temporal completeness for each chunk D_i.
- Aggregating the computed evaluations to calculate the overall temporal completeness of period P.

Different assumptions could be made in order to estimate the temporal completeness for a single time chunk D_i. Two of them are presented hereafter:

- **Assumption 1:** We consider as a reference, a uniform distribution of the measurements over time. RM_{D_i} is defined for a chunk of time D_i as:

$$RM_{D_i} = \Sigma_{j=1}^{K} n_{sj} \tag{3.8}$$

  Where K is the number of sensors, n_{sj} is the number of required measurements for sensor s_j during the time chunk D_i.

  For a sensor s_j, the number of required measurements during a chunk of time D_i is computed as:

$$n_{sj} = f_{sj} * |D_i| \tag{3.9}$$

  Where f_{sj} is the sampling rate of the sensor s_j expressed in the number of measurements per minute, and $|D_i|$ is the size of the chunk D_i expressed in minutes.

- **Assumption 2:** We consider that the measurements are distributed considering the variation of measured element levels at different times of the day, month, or year. Some physical phenomena could be highly time dependent (for example, the air pollutants levels are higher at rush hours than at other times of the day). A possible approach would be to analyze the available data to detect variation patterns by studying the trends in the existing data. The number of required measurements can then be set based on these patterns in order to compute the temporal completeness.

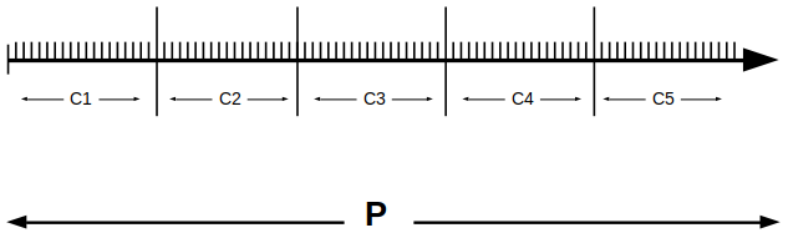

Figure 3.4: Period P divided into chunks D_i.

Temporal Completeness of a Specified Chunk of Time D_i

After dividing the time period P into time chunks, we compute the temporal completeness for each time chunk in period P.

Definition 3.4.5 (Temporal completeness)**.** Temporal completeness is the extent to which the available measurements cover a period of time P. The temporal completeness for a single time chunk D_i in P is computed as:

$$TC_i = \frac{AM_{Di}}{RM_{Di}} \tag{3.10}$$

Where AM_{Di} is the actual number of measurements in a chunk of time D_i and RM_{Di} is the required number of measurements in the chunk of time D_i.

We give an example of temporal completeness in the following.

Example 3.4.3. Given data coming from 4 sensors within a time period d, the temporal completeness over period d is the ratio of the actual number of measurements observed by all sensors within the time period d, to the required/reference number. If we assume the required number of measurements within this time period d is 2000 measurements computed by an estimation of rush hours and the traffic during this period. The actual observed numbers from the 10 sensors $s_1, \ldots, s_4 = \{$ 350, 200, 480, 400$\}$ respectively, then the temporal completeness over time period d is their sum (350 + 200 + 480 + 400)1430/5000 = 0.715.

Temporal Completeness of a Period P

The temporal completeness of a period of time P provides information about the way the available measurements are distributed over P, and how well P is covered by these measurements. It is computed by aggregating the temporal completeness values computed for all the time chunks in P.

Definition 3.4.6 (Temporal completeness of a period P)**.** Temporal completeness for a time period P can be computed as the average of all the temporal completeness values of its chunks, as shown below:

$$TC_P = \frac{\Sigma_{i=1}^{|P|} TC_i}{|P|} \tag{3.11}$$

Where $|P|$ is the number of chunks in a period of time P, and TC_i the temporal completeness of chunk D_i.

3.5 . Quality-aware data imputation for completeness improvement

To improve the completeness of data in an application context, one of the available approaches is to use data imputation techniques to generate and replace the missing values. Data imputation techniques rely on existing data measurements, from other sensors for example, or from the same sensor but at different timestamps, to replace a missing value. Some techniques impute a value based on geographically close measurements. Other techniques generate a missing value based on historical data measurements taken at the same location a year or a month ago depending on the seasonality. Mobile light low-cost sensors are prone to points of failure, and other malfunctions can take a long time to be fixed, leading to important missing chunks in the data. If we analyze the data and compute indicators based on data with low completeness, the findings could be misleading, resulting in wrong decisions taken. There are many reasons why low-cost mobile sensors could perform poorly, for example, a sensor can perform poorly due to misuses such as wrong orientation, the regular shutdown of the sensor, or a degraded battery.

Several works have focused on improving data completeness by generating the missing values using data imputation techniques for time series data coming from sensors [Yi et al., 2016], [Khayati et al., 2014], [Troyanskaya et al., 2001], but none of these works take the quality aspect into consideration. Considering quality helps data imputation techniques prioritize measurements coming from good-quality sensors over measurements coming from low-quality ones. For example, if a sensor unit s_i was showing a series of aberrant measurements between two time periods, we could choose to ignore the measurements coming from this sensor between these two time periods for any imputation task. Our goal is to enhance data imputation by taking into account the available information about the quality of the data. This quality could be either a quality value related to a single quality factor or the aggregation of quality values corresponding to several quality factors.

In this section, we discuss some existing data imputation techniques from different families explored in depth in section 2.4 in the Related Works chapter. Then, we define sensor quality and its metrics. The sensor quality is later used to extend three existing data imputation techniques ST-MVL, SVDImpute, and KNNImpute.

3.5.1 . Sensor quality metrics

Before we present our quality-aware extensions of existing data imputation techniques, we first define the notion of the quality of a sensor. Then, we give the

idea behind sensor quality and provide the metrics as measures to characterize it. We then propose a function to aggregate several measures that characterize the sensor. We consider that the resulting quality score of the sensor is a weighted average of these different normalized quality metrics. Finally, we showcase our proposed metrics and aggregation function on a real case study with sensor units measuring particulate matter.

External circumstances and conditions can impact the way a sensor performs. Various reasons could cause some sensors to perform better than others, such as the characteristics of the device itself, like the measurement acquisition rate or the technologies at the heart of this device. In addition, the performance of sensor units of the same type and coming from the same manufacturer may vary depending on the meteorological context in which the measurements were taken. For example, the indoor air quality measuring sensor, NETATMO[1] operates correctly only for an external temperature between 0°C and 50°C with a humidity level between 0% and 100%. We can therefore deduce that the quality of the data provided by this sensor when the temperature is negative, is of poor quality. The quality of the measurements can also be impacted by the way the sensor is used by its carrier. Indeed, a sensor whose battery is discharged, will provide data of lower quality than a sensor operating continuously with a high battery level. Hence, the data coming from each sensor unit can be disparate in terms of quality due to all of the aforementioned factors. These can therefore be used to determine the level of quality of a sensor.

Many external and internal circumstances can have either a significant negative or positive impact on the performance of a sensor unit. Hence, metrics are required in order to assess the performance of a sensor unit to understand the reliability of a given sensor. There are different ways of evaluating sensor quality. We can assess any of the data quality factors like completeness, accuracy, consistency, timeliness, reliability, among others. To evaluate the accuracy of the data coming from a sensor, we can compare the measurements values to data from reference devices if available. Moreover, it is possible to assess a sensor unit based on the physical features and the technologies that constitute the sensor, such as the sensitivity of the sensor to concentration changes or its capability to capture the temporal variability of the measured element as proposed by the authors of [Fishbain et al., 2017]. It is also possible to aggregate several quality metrics to assess sensor quality. However, it is not always the case that such resources are available. There are different perspectives on how to define sensor quality and many quality measures could be added, normalized, and aggregated to represent sensor quality. In our work, we assess sensor quality as the aggregation of three different quality measures described hereafter.

[1] https://www.netatmo.com/fr-fr/aircare/homecoach/specifications.

Device level metric: device-level metrics include metrics that describe the quality of the sensor at the device level. These metrics could assess some physical characteristics of the sensor. The set tool [Fishbain et al., 2017] was used by [Languille et al., 2020] to conduct static tests to do an initial evaluation of the sensor units within the project to evaluate the actual quality performance index of the sensor units available. The tool comprises metrics that test the data coming from the sensor, such as presence that accounts for the sensor's availability of a measurement at a given time, as well as some physical features, such as source analysis, which assesses the ability of the sensor to react to changes in observations within a time interval that corresponds to wind direction change. The results of the evaluation using the SET tool were used as a measure in our evaluation of the quality of our sensors.

Usage Related metric: This category of metrics evaluates the performance of the sensor after usage because metrics from this family are based on analyzing the data collected by a sensor over a certain period of time. Metrics that evaluate any quality dimension of the data lie under this category of metrics. In this category of measures, it is possible to have a metric that evaluates, for example, the data completeness, accuracy, detection of anomalies, etc. We used the sensor completeness metric proposed in subsection 3.4.1 to evaluate the usage-related performance of the sensors. We evaluate our sensors using this metric that studies the overall completeness of the measurements coming from the sensor across multiple usages of the sensor unit over a certain period of time.

Reference-Related metric: This category of metrics evaluates the data coming from the sensor compared to some reference data if available. For this metric, we compare the available dataset and the reference dataset using correlation coefficients that help detect the degree of correlation between the two datasets, and some error metrics such as the RMSE that computes the error between the values of both datasets. For example, the Pearson coefficient expresses how coherent the two datasets are without amplifying the differences. And the RMSE tells us how different these two datasets are. Hence, if the data from a sensor has a high correlation coefficient and a low RMSE at the same time, then this is most probably good-quality data because this means that the differences between the values among the two datasets are relatively low and they are coherent.

Aggregated sensor quality

In order to compute a global quality value, we propose to aggregate all of the available quality values provided by a set of quality metrics that assess different aspects of the sensor. The aggregated value is computed by a weighted average of each quality metric that is assigned a weight value. The weight value associated

with each computed quality value is assigned depending on the reliability of the computed value. For example, if the reference data are of high reliability, we assign a high weight to the values computed using the metrics comparing the data with reference data.

Assume that n quality metrics $\{qm_1, \ldots, qm_n\}$ are available to define the quality of sensor s_i. We define an aggregated quality measure for sensor quality $Q(s_k)$ that uses a weight vector $\omega = (w_1, \ldots, w_n)$ to weight the values computed using the quality metrics and compute a global quality index of the sensor s_k that summarizes the quality values computed by all the available metrics.

Definition 3.5.1 (Sensor quality)**.** Sensor quality is a global quality value assigned to a sensor at a specific time to assess its performance. The global quality value is computed as a weighted aggregate of several quality values that are computed to assess different aspects of the sensor. The aggregated value is computed as follows.

$$Q_{s_k} = \frac{\sum_{i=1}^{n} w_i \times qm_i}{n} \tag{3.12}$$

Where n is the number of quality measures, qm_i is a quality measure computed by some quality metric, s_k is the assessed sensor, and w_i is the weight assigned to quality measure qm_i.

Example 3.5.1. In our context, we applied this definition to compute the quality value of available sensors. We computed the three different metrics assessing different aspects of the sensors discussed earlier. We then assigned equal weights to the metrics computed to compute a final global aggregated value of each sensor presented in Table 3.1.
Table 3.1 shows an application of this concept on data collected by sensors in a mobile crowdsensing environment (MCS) where we present the results of the aggregation of three computed metrics from each category of measures.

Table 3.1: Example of computation of sensor quality.

Sensor Unit	F1	F2	F3	F4	F5	F6	F7	F8	F9
Sensor Quality Value	0.64	0.63	0.69	0.64	0.69	0.67	0.1	0.65	0.51

Sensor Unit	F10	F11	F12	F13	F14	F15	F16	F17	
Sensor Quality Value	0.46	0.53	0.61	0.59	0.55	0.68	0.49	0.85	

3.5.2 . Extension of ST-MVL

ST-MVL is a pattern-based approach that uses a multi-view learning algorithm to compute a weighted average of imputed values from four different techniques. The first sub-technique is the inverse distance weighting (IDW) technique which interpolates a missing value based on its spatial neighborhoods. The second sub-technique is the simple exponential smoothing (SES) that estimates the missing value based on the readings of the same sensor at other timestamps. The third sub-technique is user-based collaborative filtering (UCF) which computes similarities between users to estimate a missing value. The last sub-technique is item-based filtering (ICF) which computes the similarity between two timestamps to estimate a missing value. Finally, ST-MVL integrates the predictions of the views of the four aforementioned sub-techniques to generate a final value through a multi-view learning algorithm that learns the weights to assign to each one of the four sub-techniques based on existing data. Finally, it computes a weighted average of the four sub-techniques estimations to generate a missing value.

As presented in the Related Works chapter, suppose the value estimated by IDW is denoted by $\hat{v}_{gs}$, the value estimated by SES is denoted by $\hat{v}_{gt}$, the value estimated by UCF is denoted by $\hat{v}_{ls}$, and the value estimated by ICF is denoted by $\hat{v}_{lt}$. The weights assigned to the techniques are represented by w_i, $i \in \{1, 2, 3, 4\}$ corresponding to each sub-technique respectively. The final generated value by ST-MVL $\hat{v}_{mvl}$ of a single missing value v is computed as follows.

$$\hat{v}_{mvl} = w_1 * \hat{v}_{gs} + w_2 * \hat{v}_{gt} + w_3 * \hat{v}_{ls} + w_4 * \hat{v}_{lt} + b \tag{3.13}$$

where b is a residual and w_i (i = 1,2,3,4) is the weight assigned to each sub-technique respectively.

As part of our proposition, we extend the existing algorithm by taking into account the sensor quality, which gives insight into the quality of the sensor presented in subsection 3.5.1. We extend these techniques with quality of the sensors as evaluated with the metrics proposed earlier to show that considering the quality of the measurements improves the imputation of a missing value.

Extending IDW

Inverse distance weighting (IDW) is a statistical model used to interpolate missing values at a given timestamp, based on the spatially closest sensor readings. It assigns weights to the measurements according to their spatial distance from the target/missing measurement. The technique is discussed in further detail in subsection 2.4.2. To extend this technique with quality, we compute a weighted average of the measurements weighted by their distance to the target measurement, over the sum of both weights (i.e., distance and quality) to amplify the magnitude of measurements with higher quality over others with lower quality. The intuition

here is to consider sensor quality in a similar way to the distance by this algorithm. The technique considers distance a weight that gives higher importance to closer measurements and lower importance to further ones. Likewise, our extension assigns higher-quality measurements a greater importance than lower-quality ones. Extended IDW estimates a missing value v_i according to the following formula.

$$\hat{v}_{gs} = \frac{\sum_{i=1}^{m} v_i * d_i^{-\alpha} * q_{s_i}}{\sum_{i=1}^{m} d_i^{-\alpha} * q_{s_i}} \tag{3.14}$$

Where α is a positive power parameter that controls the decay rate of a sensor's weight by distance, and q_{s_i} is the quality of sensor s_i.

Extending SES

Simple exponential smoothing (SES) is a technique that estimates the missing value based on the temporally adjacent measurements of the same sensor. It assigns higher weights to measurements that are temporally closer to the target missing measurement than the others. The technique is discussed in detail in subsection 2.4.2. SES is extended by introducing the sensor quality to the product of the measurements that are already weighted by their temporal closeness to the target measurement, by the quality of the sensor taking the measurement. The intuition here is similar to that in Equation 3.5.2, to assign a higher weight to measurements with higher sensor quality in a similar way it is done by the technique to recent measurements.
Extended SES estimates a missing value v_j according to the following.

$$\hat{v}_{gt} = \frac{\sum_{j=1}^{n} v_j * \beta * (1-\beta)^{t_{j-1}} * q_{s_i}}{\sum_{j=1}^{n} \beta * (1-\beta)^{t_{j-1}} * q_{s_i}} \tag{3.15}$$

Where β is a smoothing parameter, $\beta * (1-\beta)^{t-1}$ is used to give a higher weight to recent readings than distance ones, and q_{s_i} is the quality of sensor s_i.

Extending UCF and ICF

User-based, and item-based collaborative filtering are two techniques wildly used in recommender systems [Li et al., 2009]. The idea behind user-based collaborative filtering (UCF) is that similar sensors produce similar measurements. Hence, it uses the similarity between two sensors as a weight to prioritize some measurements over others. To extend UCF with sensor quality, a measurement is weighted by the quality of the sensor that took the measurement that is later in the UCF weighted by the similarity between the sensor taking it and the target sensor. This is meant

to give a higher importance to measurements with higher quality. Extended UCF estimates a missing value based on Equation 3.16.

$$\hat{v}_{ls} = \frac{\sum_{i=1}^{m} v_i * sim_i * q_{s_i}}{\sum_{i=1}^{m} sim_i * q_{s_i}} \tag{3.16}$$

Where sim_i is the similarity between the sensor that took v_i and the target sensor of the missing measurement, and q_{s_i} is the quality of the sensor s_i. Similarly, item-based collaborative filtering assumes identical timestamps must have similar measurements. Hence, it uses the similarity between two timestamps as a weight to prioritize some measurements over others. To extend ICF with sensor quality, we weight a measurement that is weighted by the similarity between the timestamp at which it was taken and the timestamp at-which the target measurement was taken, by the quality of the sensor taking the measurement. Extended ICF estimates a missing value v_i as shown in Equation 3.17.

Similarly to extended IDW and SES, the intuition behind this proposed extension with sensor quality is to weight the computed values by the quality of the measuring sensor such that we dedicate a higher significance to measurements with higher quality than others with lower quality.

$$\hat{v}_{lt} = \frac{\sum_{j=j_1}^{j_2} v_j * sim_j * q_{s_i}}{\sum_{j=j_1}^{j_2} sim_j * q_{s_i}} \tag{3.17}$$

Where sim_j is the similarity between the timestamp at which v_i was taken and the timestamp of the missing measurement, and q_{s_i} is the quality of sensor s_i.

3.5.3 . Filtering of SVDImpute

SVDImpute is a technique that employs the Singular Value Decomposition (SVD) to obtain the principal components of the matrix A that comprises the data measurements from sensors at their collected timestamps, and regresses these principal components against the existing data measurements from the target sensor. The technique is explained further in subsection 2.4.1. We propose two possible extensions to SVDImpute in order to take into account sensor quality. One considers only a certain percentage of sensors having the highest quality. For example, data coming from only top 70% quality sensors are considered in the imputation. The second takes sensors having a quality score above a predefined threshold where only the data measurements with a quality value above a certain threshold are taken into account during the imputation.

The proposed extension of SVDImpute comprises a preprocessing step that filters the data that the technique is going to use in the imputation according to quality criteria. The first proposed extension considers only the data of the percentage *p* of the sensors having the highest quality score. For example, if we set $p = 70\%$ and if we have data from 10 sensors, SVDImpute will consider only the data from the seven sensors with the higher quality scores. The data from the three remaining sensors with lower quality scores will be discarded.
The second proposed extension is similar to the first one, but instead of considering the percentage *p* of sensors having the highest quality score, it considers sensors having a quality score above a threshold γ, where γ ranges between 0 and 1. In this proposition, we are only interested in the data from the sensors that have a quality score above a predefined threshold. If we consider the previous example and assume our threshold is 70% and only four sensors out of 10 have a quality score above 70%. In this case, we only consider the data from these four sensors and discard the remaining ones.

Assume we have data measurements from sensors $s_i = \{a_1, \ldots, a_m\}$ where each measurement is taken at timestamp t_j. A data matrix $A_{m,n}$ is a matrix comprising data from m sensors having measurements at n timestamps. We represent a filter $K_{m,m}$ by a matrix of ones with values of zero at the locations that filter out the rows that need to be discarded.

For both extensions, we filter out the data of the unwanted sensors by multiplying the data matrix $A_{m,n}$ by a filter $K_{m,m}$ that nullifies the rows of the unwanted sensors. Hence, the resulting matrix $A'_{m,n}$ only contains the rows of data from the selected sensors. For example, if matrix A is of size 4×5 as shown below, representing data from 4 sensors over 5 timestamps, and suppose only sensors s_2 and s_3 have a quality score above the considered threshold. Then, the matrix $A'_{m,n}$ will be computed as follows:

$$A'_{m,n} = K_{m,m} \times A_{m,n} \tag{3.18}$$

$$\underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{K_{m,m}} \times \underbrace{\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{array} \begin{array}{c} \begin{array}{ccccc} t_1 & t_2 & t_3 & t_4 & t_5 \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{pmatrix} \end{array}}_{A_{m,n}} = \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{A'_{m,n}}$$

3.5.4 . Extension of KNNImpute

KNNImpute is a technique that generates a missing value at a specific timestamp based on the measurements from neighboring sensors at the same timestamp. It chooses the k-nearest sensors to be used for the imputation of the missing value. The technique is explained further in subsection 2.4.1.

To extend KNNImpute, we weight the measurements with the quality score of the measuring sensors. This will result in giving more importance to the measurements coming from good-quality sensors over those from poor-quality ones.
Consider a set of sensors S = {s_1,..., s_n} where each sensor s_i has a quality score q_{s_i}. Assume sensor s_t has a missing measurement v_j at timestamp t_j. The imputed value $\hat{v}_j$ generated by the extended version of KNNImpute is defined by:

$$\hat{v}_j = \frac{\sum_{l=1}^{k} v_{lj} * d_{s_l,s_t} * q_{s_l}}{\sum_{l=1}^{k} d_{s_l,s_t} * q_{s_l}} \tag{3.19}$$

Where *k* is the number of neighboring sensors, d_{s_l,s_t} is the Euclidean distance between sensor s_l and target sensor s_t, and v_{lj} is the measurement value of sensor s_l at timestamp t_j.
In the following two sections, we will present the evaluations of the quality metrics proposed to assess each of the proposed completeness factors, and the evaluations of the three extended techniques using quality on time series data taken within the context of a mobile crowdsensing environment.

3.6 . Data completeness metrics evaluation

This section presents our experiments on evaluating the three defined data completeness factors: sensor completeness, spatial completeness, and temporal completeness.

3.6.1 . Experimental setup and dataset

We have conducted our experiments within the context of the Polluscope project[2] [Brahem et al., 2021], [Languille et al., 2020] that aims to quantify human exposure to air pollutants within a mobile crowdsensing environment. A sensor s_i in a mobile crowdsensing environment collects a set of measurements $X_n = \{v_{i1}, v_{i2}, \ldots v_{in}\}$, where v_{ij} is a measurement vector taken by sensor s_i at timestamp t_j, and measuring a certain physical element.

In our context, the dataset comprises a set D of time series, $D = \{X_1, X_2, ..., X_n\}$, acquired by a set of sensors S = {s_1,s_2, ..., s_n}.

[2] http://polluscope.uvsq.fr

Each series X_i is composed of data measurements v_{ij}. Each sensor collects measurement values of a specific air pollutant alongside the timestamp and other features. The air pollutants collected in the Polluscope project are nitrogen dioxide NO_2, particulate matter with three different diameters: $PM_{1.0}$, $PM_{2.5}$, $PM_{10.0}$, and black carbon BC.

As defined in subsection 3.3.1, each measurement vector v_{ij} has the following features. The measurement value which is the level of the measured pollutant $v \in \mathbb{R}$, the timestamp t_j that indicates the time at which the measurement was taken, a pair (lat,lon) that indicates the spatial geographic coordinates where the measurement was taken, and s_i that represents the sensor unit that took the measurement. Hence, the measurement vector v_{ij} is represented by $v_{ij} = (v, t_j, (lat, lon), s_i)$ which is the set of all the features that characterize this measurement.

Setup of the experiments

We conduct our experiments to evaluate the three factors of data completeness: sensor, temporal, and spatial completeness presented in section 3.4 using our proposed metrics over a real dataset collected within the context of the Polluscope project [Brahem et al., 2021], [Languille et al., 2020] with a total size of 5 856 540 data measurements measuring the following air pollutants: NO_2, BC, $PM_{1.0}$, $PM_{2.5}$, $PM_{10.0}$. The data was collected over two distinct campaigns. 1 627 487 data measurements were taken during the first campaign, and 4 229 053 measurements were taken during the second campaign. The campaigns took place in île-de-france. Each campaign has a start and an end date. People participating in a campaign are called users. The users carry several sensors of different types measuring different pollutants, which they carry around their daily lives and usual routines, for a predefined time duration. The measured pollutants are particulate matter (PM10, PM2.5, and PM1.0), NO2, or black carbon (BC). A sensor unit may be used twice within the same campaign by two different users at different time periods. Each measurement is associated with a timestamp and a geographical location.

For the sensor completeness experiments, we have considered a subset of this data taken by one sensor unit measuring NO_2 over two campaigns. We have extracted 21 398 NO_2 measurements from the first campaign, and 38 834 measurements from the second campaign for the sensor completeness experiments.

For the spatial completeness experiments, we have considered all the datasets from campaigns one and two to evaluate the spatial completeness per pollutant.

Finally, to evaluate the temporal completeness we have extracted a subset of the initial dataset that is measuring three pollutants $PM_{2.5}$, BC, and NO_2 over

both campaigns. The subset contains 582 506 measurements from campaign one and 1 378 497 measurements from campaign two.

3.6.2 . Evaluating sensor completeness

For this experiment, we have selected one sensor unit that measures the NO_2 pollutant. The experiments are performed on the NO_2 pollutant measurements obtained both in campaigns 1 and 2 by this sensor. Hence the sensor completeness of the selected sensor unit measuring NO_2 is studied over the period of the two separately, then studied over the 2 campaigns combined. The sensor completeness value was 58.66% and 59.92% for campaigns 1 and 2, respectively. Generally, the sensor has performed slightly better on the second campaign than the first. However, values achieved by the sensor over the two campaigns are in the same range. Table 3.2 shows the sensor completeness of the selected sensor in all its usages during campaign 2.

Table 3.2: Completeness of a sensor measuring NO2 for all its usages during campaign 2.

Usage Nb	Sen-Comp
1	37.65%
2	77.3%
3	70.20%
4	28.55%
5	87.89%

3.6.3 . Evaluating spatial completeness

We conducted the experiments to evaluate spatial completeness on each of the following air pollutants: $PM_{1.0}$, $PM_{2.5}$, PM_{10}, NO_2, and BC over both campaigns 1 and 2. The evaluations are done over a manually pre-selected area in Paris.

Campaign 1 has less collected data than campaign 2. We first compute spatial completeness as described in subsection 3.4.2 for every pollutant and for each of the sensors used in this campaign, and then we compute an average of all the sensors to get the total spatial completeness. The fact that campaign 1 has less collected data than campaign 2 should be taken into consideration when analyzing spatial completeness because it means that with more data in campaign 2, there is the possibility of wider spatial coverage. Table 3.3 shows the spatial completeness values computed for campaigns 1 and 2. The values of the different pollutants were more or less in the same range, and significantly lower in the first campaign. However, we notice an improvement in the second campaign, and the spatial completeness is still fairly low for all of the measured pollutants.

Table 3.3: Spatial Completeness of all pollutants during campaigns 1 and 2.

Pollutants	SC Campaign 1	SC Campaign 2
PM1.0	15.10%	33.02%
PM2.5	15.10%	33.02%
PM10	15.10%	33.02%
NO_2	18.17%	35.15%
BC	20.38%	34.99%
Temperature	15.10%	32.91%
Humidity	15.10%	32.91%
Pressure	15.10%	33.024%

3.6.4 . Evaluating temporal completeness

Over the two campaigns 1 and 2, we have also evaluated temporal completeness for each of the pollutants *PM2.5*, NO_2 and *BC*. Table 3.4 shows the temporal completeness of the three pollutants *PM2.5*, NO_2 and *BC* over campaigns 1 and 2. The results of both campaigns 1 and 2 appear to be in the same range for the

Table 3.4: Aggregated total average of Temporal Completeness of all pollutants during each sensing campaign.

Pollutants	TC Campaign 1	TC Campaign 2
PM2.5	7.75%	42.23%
NO_2	60.91%	63.66%
BC	68.53%	59.49%

two pollutants NO_2 and BC, around 60% which is relatively high. Aside from the one extremely poor temporal completeness value achieved in the first campaign for $PM_{2.5}$, the remaining temporal completeness values achieved are around 60%.

3.6.5 . Discussion

The purpose of the experiments was to analyze the quality of the collected data using the proposed data completeness factors and their associated metrics. The first assessed factor was sensor completeness, where we selected one sensor unit to evaluate over two campaigns. The disparate values of the achieved sensor completeness by the same sensor unit show that our metrics succeeded at capturing the real faulty nature of the sensors at times or the poor usage of the sensor by the carrier. The same sensor unit performed very well, 87.89% at times and poorly at other times, 27.55% most certainly due to

conditional circumstances that happened to the sensor unit around the time of the measurements. This means that a sensor unit performing poorly at the moment could later perform with better conditions and/or better usage by the carrier.

The second assessed facet was spatial completeness where we manually selected the designated area to study in the heart of Paris. We evaluated all of the pollutants $PM_{1.0}$, $PM_{2.5}$, PM_{10}, NO_2, and BC over two campaigns 1 and 2. The studied area appears to have better spatial completeness in the second campaign than in the first.

Given the fact that more users have participated in the second campaign than the first one, this can explain the result because it means less possibilities of spatial coverage. We noticed the existence of patterns in the trajectories followed by the users (e.g. mostly back and forth from home to work) which explains the low spatial completeness covered by the participants in general. This shows that our metric for spatial completeness facets proved useful because the computed value reflects the reality of the spatial completeness of this data.

Finally, the experiments of the temporal completeness factor were conducted for the duration of the two campaigns of the three pollutants $PM_{2.5}$, NO_2, and BC.

The temporal completeness achieved in campaign 1 for $PM_{2.5}$ is significantly low because several $PM_{2.5}$ sensors had a software defect during the first campaign that made them lose vast chunks of their collected data. Otherwise, the remaining achieved temporal completeness for NO_2 and BC is better because we have noticed that these sensors functioned more steadily because they did not lose their collected data due to malfunctions with sensors measuring PM. This implies that the metric for temporal completeness correctly reflects the reality of the completeness of the data captured by the different sensor types.

3.7 . Evaluation of our quality-aware data imputation approach

In this section, we present our experiments and the results achieved with the proposed quality based extensions of the three data imputation approaches: ST-MVL, SVDImpute, and KNNImpute. We first describe our experimental setup, and then we show the evaluation metrics that we will use throughout the experiments. Finally, we show the main results achieved for all of the three techniques. The rest of the experiments presenting more quality thresholds and percentages of SVDImpute, and results of the sub-techniques of ST-MVL will be found in the appendix in Appendix A.

3.7.1 . Experimental Setup

We have taken all the available data related to one participant which initially represents 7 700 measurements. We have evaluated the performance of our extensions on two datasets:

1. Dataset 1: represents a partition of the data where we have 2 other participants geographically close to the target participant. This is because the techniques rely on the spatial distance between the missing measurement and the other available measurements.

2. Dataset 2: represents the entire set of data for our participant.

During our experiments, we test the performance of our proposals by first assuming every measurement in the subset is missing. We recall that the sensor quality we test is composed of quality values resulting from three categories of quality metrics. The first computed measure is from the device-level category where the set tool [Fishbain et al., 2017] has been used by the work of [Languille et al., 2020] to evaluate the sensors and compute an index that is denoted by the IPI index throughout our experiments. The second measure is the sensor completeness computed in our work in [Mehanna et al., 2020], subsection 3.4.1, and finally, the third measure computed using correlation coefficient and RMSE from the comparison-related category of metrics.

We then study the performance of our extensions on ten different sets of weight configurations represented by the percentages set [w_1, w_2, w_3]. Each one of the three weights w_i $i \in \{1, 2, 3\}$ corresponds to a quality value computed using a quality metric from each of the categories discussed earlier in subsection 3.5.1 in order to assess the quality of a sensor. The value of the weight w_1 is the weight percentage assigned to the quality value computed using a device-level metric, the value of the weight w_2 is the weight percentage assigned to the quality value computed using a usage-related metric, and finally the weight w_3 is the weight percentage assigned to the quality vallue computed using a comparison-to-reference metric. We select ten weight configurations to test our extensions with, as shown in Table 3.5. The ten weight configurations correspond to different settings: (i) some where all three categories of quality factors are assigned a similar weight, (ii) others where only one of the three factors is considered, and (iii) others where the weights are distributed while giving one factor a higher weight than the other two. These ten weight configurations are used to determine the impact of each quality factor on the results of our extensions.

The quality of the sensor is instantiated in our experiments with the following data evaluated by quality metrics from each category presented in subsection 3.5.1.

- The SET tool is a device-level tool that comprises metrics used to compute the IPI index that represents a global quality value that assesses different aspects of sensor devices.

- The metric proposed to measure sensor completeness in subsection 3.4.1 is a metric that we use in our experiments as a usage-related measure characterizing one aspect of the quality of the sensor.
- The comparison with reference data is the third metric used in our experiments where we compare the data collected by our sensors with data from reference devices provided by AirParif.

Table 3.5: The distribution of the 10 weight configurations over the 3 measures of sensor quality.

IPI index%	Sensor Completeness%	Correlations with Reference%
33	33	34
100	0	0
0	100	0
0	0	100
60	20	20
20	60	20
20	20	60
40	40	20
40	20	40
20	40	40

3.7.2 . Evaluation Metrics

To assess the performance of the proposed extension to the existing data imputation approaches, we evaluate the error using the RMSE metric to compute the error between a generated series and the actual values. This metric can be computed between the actual data and the imputed data by the basic technique as well as between the actual data and the imputed data by the extended techniques. We then compare the error metric resulting from the basic technique and the error metric resulting from the extended techniques.

We also compute other performance indicators: the proportion of measurements improved, the proportion unchanged and the proportion worsened. The proportion of measurements improved is the proportion of measurements that were more accurately imputed by our extensions than by the basic approach. Those worsened are the measurements that were less accurately imputed by our extensions that by the basic approach. Finally, the unchanged are the ones for which both the basic and the extended approach provided the same imputed value.

RMSE Error Metric

RMSE measures the quality of an estimator. To quantify the error of the estimation models, we compute the root mean squared error (RMSE) metric between a vector of estimated/predicted values $\hat{v_{ij}}$ and a vector of the actual/observed values v_{ij}. The RMSE could be computed for the estimation of the baseline approaches as well as the estimation from the extended approaches proposed in this chapter in section 3.5. With this, we can compare the error metric RMSE between the estimations from the baseline approach such as SVDImpute, ST-MVL or KNNImpute vs. the extended versions of the aforementioned approaches.

Statistical Indicators

The proportion of improved measurements shows the number of measurements improved using the quality-based data imputation compared to the baseline approach. The proportion of worsened measurements shows the number of those that had a more accurate estimation of the missing measurements using the baseline approach. Finally, the proportion of unchanged measurements represents those with the same imputed value using both approaches. The intuition behind this is that the further the imputed value is from the actual one, the worse the imputation approach. Hence, the imputation of a missing measurement has been improved using our extension if the absolute value of the difference between the actual measurement and the imputed measurement using the extensions has to be smaller than the absolute value of the difference between the actual measurement and the imputed measurement by the baseline approach.

Assume an actual measurement is denoted by v_{ij}, an imputed measurement generated by our extensions is denoted by $\hat{v}^e_{ij}$, and an imputed measurement generated by the baseline approach is denoted by $\hat{v}_{ij}$.

$$|v_{ij} - \hat{v}^e_{ij}| < |v_{ij} - \hat{v}_{ij}| \tag{3.20}$$

In equation 3.20, $|v_{ij} - \hat{v}^e_{ij}|$ is the absolute difference between the actual value and the imputed value generated by the extended approach, and $|v_{ij} - \hat{v}_{ij}|$ is the absolute difference between the actual value and the imputed value generated by the baseline approach.

If equation 3.20 holds, this means that the extended approach was more accurate than the baseline approach. This implies that the imputed extended value here is closer to the actual one from the value imputed by the baseline approach.

However, if the equation 3.20 does not hold, and $|v_{ij} - \hat{v}^e_{ij}| = |v_{ij} - \hat{v}_{ij}|$, this means that both the baseline/baseline and the extended approaches both imputed the same value. Therefore, the proposed extension did not improve the imputation.

Finally, if $|v_{ij} - \hat{v}^e_{ij}| > |v_{ij} - \hat{v}_{ij}|$, this means that the imputed value generated

by the baseline approach was closer to the actual value than that of the value imputed by our extended approach. In this case, the imputation is not improved using the extended approach.

3.7.3 . Evaluation of extended ST-MVL

We have tested our extension of ST-MVL on the two datasets presented in subsection 3.7.1. First, we have tested our proposal with a subset of the data related to some selected users. The participants for this dataset were selected if they were close to at least two other participants. We have set this condition because the studied techniques rely on the spatial distance between two measurements. Then, we have tested our extension on another dataset which comprises the entire data collected during the usage of the selected participant including times where the participant was at proximity and the times where the participant was not at proximity with other users.

Results on Dataset 1

Table 3.6: Results of ST-MVL for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	46%	20%	34%	1.11
$[100, 0, 0]$	30%	7%	63%	1.11
$[0, 100, 0]$	51%	20%	29%	1.11
$[0, 0, 100]$	47%	37%	16%	1.11
$[60, 20, 20]$	40%	11%	49%	1.11
$[20, 60, 20]$	47%	17%	36%	1.11
$[20, 20, 60]$	45%	27%	28%	1.11
$[40, 40, 20]$	43%	15%	42%	1.11
$[40, 20, 40]$	44%	21%	35%	1.11
$[20, 40, 40]$	47%	22%	31%	1.11

Table 3.6 shows the results of the evaluation of the proposed extension of ST-MVL on the first dataset. The best improvement result was 51% achieved using the weight configuration $[0, 100, 0]$. 20% of the measurements computed using this weight configuration were worsened with a very small RMSE error value 1.11. 29% of the measurements remained unchanged with this weight configuration.

The weight configuration $[100, 0, 0]$ had the highest unchanged percentage 63%, showing again, that the IPI index facet alone, does not have a huge impact in improving or worsening the imputation. It also showed 30% improvement of the imputations and 7% worsening with 1.11 RMSE error value. The RMSE

error value was constant at 1.11 for all the different weight configurations of this experiment.

Generally, more measurements were improved than worsened for this dataset for ST-MVL. The percentage of worsened measurements are generally low except for the weight configuration $[0, 0, 100]$ which produced 37% worsened measurements.

Results on Dataset 2

Table 3.7: Results of ST-MVL for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	48%	43%	9%	19
$[100, 0, 0]$	52%	40%	8%	19.1
$[0, 100, 0]$	53%	30%	17%	19
$[0, 0, 100]$	49%	43%	8%	19.2
$[60, 20, 20]$	49%	31%	20%	19
$[20, 60, 20]$	48%	33%	19	18.9
$[20, 20, 60]$	52%	41%	7%	19.1
$[40, 40, 20]$	49%	42%	9%	19
$[40, 20, 40]$	49%	43%	8%	19.1
$[20, 40, 40]$	47%	35%	18%	19

Table 3.7 shows the results of the evaluation of the extension of ST-MVL on the whole dataset of the designated participant. The improvement percentage range of this dataset shows similar results for the different weight configurations but with tenuous differences. 53% is the highest achieved improvement result for the weight configuration $[0, 100, 0]$ that considers only the sensor completeness factor.

The percentages of measurements worsened for this experiment are similar to the percentages of the measurements improved, this shows that our extension did not work out very well for this technique and dataset. However, the RMSE error was not high, so this could mean that even though some measurements were worsened, the difference between the extended and the baseline was not high.

Analysis of the achieved results

There are more improved measurements in the imputation of the measurements than worsened in both datasets for the experiments performed using the extension of ST-MVL with quality-related information. The improvement results were almost the same among the two datasets with the best improvement

achieved with the $[0, 100, 0]$ weight configuration. The proportion of worsened measurements was higher for dataset 2 than for dataset 1. There is a significant difference in the percentages of measurements unchanged between the two datasets. This indicates that the extension helped improve the quality of the imputation by the technique when there were other users at a close distance.

The RMSE is higher for dataset 2 than for dataset 1. This might be due to the fact that there is a higher possibility for errors in 7 700 measurements than in 480 measurements. Another reason for the higher error rate is that the inverse distance weighting sub-technique of ST-MVL relies on imputing missing values according to neighboring sensors at proximity.

3.7.4 . Evaluation of SVDImpute after filtering data

We present in this subsection the results of the evaluation of our two proposed filtering methods of SVDImpute on the two datasets presented earlier. The first filtering method of SVDImpute includes data that have a quality above a certain threshold. For this filtering method, we test the quality thresholds 0.45, 0.55, 0.65, and 0.75 The second filtering method includes data that are among a certain value of the top quality data. For this filtering method, we test the k percentages 40%, 50%, 70%, and 90%.

Results on dataset 1 for filtering using quality-above-threshold

Table 3.8: Results of Quality above threshold 0.45 for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	38%	23%	39%	28
$[100, 0, 0]$	27%	13%	63%	12
$[0, 100, 0]$	62%	24%	14%	5.2
$[0, 0, 100]$	43%	41%	16%	19.9
$[60, 20, 20]$	26%	15%	59%	9.4
$[20, 60, 20]$	38%	26%	36%	24.4
$[20, 20, 60]$	36%	45%	19%	19.3
$[40, 40, 20]$	41%	22%	37%	8.7
$[40, 20, 40]$	38%	30%	32%	14.3
$[20, 40, 40]$	42%	37%	21%	17.1

Table 3.8 shows the results of the filtering method of SVDImpute considering only sensors above a threshold. In the experiment, we have set the threshold to 0.45. The quality threshold 0.45 allows more measurements, including those with quality below 50% (i.e 0.5), to participate in the generation of the missing values. This could result in lower quality imputations because data measurements from

sensors with lower quality are introduced to the imputation.

The weight configuration [0,100,0], which only considers sensor completeness facet to compute the sensor quality index, shows the best improvement performance. It shows 62% of improved values in the generation of missing values, meaning that sensor completeness alone achieves a significant improvement of data imputation despite including sensors with a quality value below 0.5. It also shows 14% of unchanged measurement values and 24% of worsened values with small RMSE values of 5.2.

The weight configuration [100,0,0] shows a relatively high percentage of unchanged measurement values: 63%, which means that the IPI index, which is the quality facet computed compared to reference data at the beginning of the campaign, almost does not have an impact on the quality of the imputation, because despite it being enriched with sensor quality information, the imputation performance is mostly the same as with and without the extensions. This is also proven with the weight configuration [60,20,20] that results in 59% of unchanged imputed measurements.

The weight configuration [20,20,60] produced the worst performance for sensors with a quality measure above the threshold 0.45. It showed 45% of measurements worsened with an error metric 19.3, and only 36% improved. More measurements were worsened than improved with this configuration. In brief, this means that giving a high weight to the comparison with the reference data while keeping the other two facets mainly worsens the performance of the technique. This is also shown with the weight configuration [0,0,100] producing 41% worsened generated missing measurements.

Results on dataset 2 for filtering using quality-above-threshold

Table 3.9 shows the results of the filtering method of SVDImpute considering only sensors above a 0.65 threshold on dataset 2. The best results achieved are when the weight was distributed on all the quality factors with the value of the IPI index having a slightly smaller weight than the rest [20,40,40]. The improvement with this weight configuration has a value of 63%, 33% of the measurements worsened and 5% unchanged. The two other set of weights $[20, 60, 20]$ and $[33, 33, 34]$ also show a similar performance with 62% improved measurements and 33% of them worsened.

The set of weights $[100, 0, 0]$ shows the highest unchanged value of 62% and only 26% improved values. This indicates that the IPI index alone does not have a huge impact on the imputation techniques. For the quality above 0.65 threshold, the results show that either distributing the weights over all the factors or

Table 3.9: Results of Quality above threshold 0.65 for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	62%	33%	5%	9.1
[100, 0, 0]	26%	12%	62%	2.1
[0, 100, 0]	54%	39%	7%	6.7
[0, 0, 100]	54%	39%	7%	10.8
[60, 20, 20]	53%	34%	13%	10
[20, 60, 20]	62%	33%	5%	9.3
[20, 20, 60]	54%	39%	7%	11.2
[40, 40, 20]	59%	34%	7%	6.8
[40, 20, 40]	53%	39%	8%	11.2
[20, 40, 40]	63%	33%	4%	9.1

giving a greater weight to sensor completeness than the others, produce the best improvement results.

Conclusions on Quality-above-threshold filtering

For dataset 1, the following are our conclusions on the quality above threshold filtering to SVDImpute. The quality threshold 0.45 and 0.55 results also showed that giving the sensor completeness the greater weight in combination with the others show the best improvement results. As with the 2 experiments on quality thresholds above 0.45 and 0.55, the quality threshold 0.65 also shows that giving sensor completeness the greater weight. However, when in combination with the other two factors, it shows the best improvement performance. This indicates that sensor completeness facet plays a major role in determining the quality of the sensor.

RMSE is an error metric that amplifies the aberrant outliers, so one outlier measurement with a very high difference can hugely amplify the value of RMSE. So, sometimes when the RMSE value is very high, it could be due to several outlier measurements in the studied sensors.

As for the dataset 2, different performances of the weight configurations were observed. Unlike for dataset 1 where we have at least 3 sensors at a very close proximity to each other, experiments on dataset 2 show that giving greater weight to the sensor completeness alone, does not produce the best results. It produced the highest percentage of unchanged for quality above 0.45 threshold, meaning that it had an insignificant impact on the performance of the technique. However, it had good impact improving the imputation for quality above 0.65 threshold. In terms of improvement, the results achieved on the dataset 1 are better than those achieved on dataset 2. We generalize that for dataset 2 in this filtering method,

the improvement percentage is always higher than that worsened. The 0.65 quality threshold shows better results for dataset 2 than the 0.45 and 0.55 thresholds.

Results on dataset 1 for filtering using 40% top-quality sensors

Table 3.10: Results of Top 40% sensors for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	63%	28%	9%	8.8
$[100, 0, 0]$	56%	37%	7%	3.8
$[0, 100, 0]$	71%	21%	8%	2.7
$[0, 0, 100]$	63%	30%	7%	7.8
$[60, 20, 20]$	60%	31%	9%	9.5
$[20, 60, 20]$	61%	29%	10%	4.3
$[20, 20, 60]$	60%	33%	7%	9.6
$[40, 40, 20]$	64%	27%	9%	3.9
$[40, 20, 40]$	62%	31%	7%	7.7
$[20, 40, 40]$	61%	30%	9%	9.5

Table 3.10 shows the results of the filtering method of SVDImpute considering only top 40% sensors on dataset 1. This experiment shows the best results achieved among our filtering methods for SVDImpute. The weights configuration set $[0, 100, 0]$ shows 71% improved values, which is the best result achieved for the filtering methods for SVDImpute. It also shows 21% worsened values and 8% unchanged with this configuration. Beside the weight set configuration $[0, 100, 0]$ being the set with the best results, the other results also performed very well with above 60% of improved values.

The weight configuration set $[100, 0, 0]$ is the only configuration with improved values below 60%. It also showed the highest worsened values of 37%. This may mean that considering only the IPI index does not achieve the best results. The unchanged values proportion among the different configurations were almost the same.

Results on dataset 2 of filtering using 70% top-quality sensors

Table 3.11 shows the result performance of applying a filtering on the data before applying SVDImpute considering only the top 70% sensors on dataset 2. The weight configuration $[33, 33, 34]$ has no results for this experiment because there were not enough data from the top 70% sensors with this distribution set of the weights to do the imputation by the extended approach. The best

Table 3.11: Results of Top 70% sensors for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	-%	-%	-%	-
$[100, 0, 0]$	48%	25%	27%	62533496
$[0, 100, 0]$	50%	18%	32%	218179
$[0, 0, 100]$	48%	21%	31%	27.1
$[60, 20, 20]$	49%	22%	19%	275749
$[20, 60, 20]$	49%	19%	32%	25.3
$[20, 20, 60]$	48%	21%	31%	6165529
$[40, 40, 20]$	56%	19%	25%	25.9
$[40, 20, 40]$	59%	19%	22%	41.6
$[20, 40, 40]$	62%	18%	20%	35.7

improvement performance was achieved with the weight configuration $[20, 40, 40]$ with 62% improved values, 18% worsened and 20% unchanged. The improvement performance of the other weight configurations ranges between 48% and 59%.

There are more unchanged measurements than worsened for this experiment. Both weight configurations $[0, 100, 0]$ and $[20, 60, 20]$ show the highest unchanged measurements percentage. The RMSE error value of most of the weight configurations is high indicating that the filtering with this setting can generate noise peaks at times.

Conclusions on top k% top-quality sensors filtering

For dataset 1, the following are our conclusions on the top 40%, 50%, 70% and 90% sensors filtering methods of SVDImpute. The top 40% and 50% quality sensors experiments showed very good results for the improvement of measurements on all the different weight configurations tested with $[0, 100, 0]$ being the best for top 40% quality sensor. For the 40% top quality sensors experiment, results showed that the set configuration $[100, 0, 0]$ performed the worst in terms of improvement with a value below 60%. However, when considering 50% of the top performing sensors, this weight configuration still performs less than the others, but we also observe that the weight configuration $[0, 0, 100]$ performs worse than the others. This means that as we add more sensors with lower quality, considering the comparison with reference data measure alone performs poorly.

When considering one factor only, we achieved both best and worst performances in terms of improvement, sensor completeness factor achieves the best results while the other two factors alone lead to the worst results. The improvement percentages for top 40% and top 50% experiments are in the same

range which is different from both top 70% and top 90% that lie in the same ranges as well. The results of the top 40% and top 50% are better than those of the latter. This proves that considering more data from low quality sensors leads to less accurate imputation results. Also, as we include top 70% and top 90% data, the values of unchanged measurements increase compared to those of top 40% and top 50%. This conclusion is plausible given that the baseline approach is practically considering 100% of the existing sensors measurements.

As for dataset 2, different performances were observed. In this experiment dataset, the more sensors we add from 40% to 50%, then to 70%, and finally to 90%, the higher the values of unchanged measurements. The consecutive differences are tenuous, yet, on average, the unchanged average values were increasing as we increased the percentage of included top-quality sensors. The best improvement result was 62%, achieved with the experiment considering top 70% quality sensors. This setting was better than the other percentages of top quality sensors studied. In general, the results achieved with this dataset are relatively good but not the best compared to the others. Compared to dataset 1, experiments on dataset 1 show significantly better results. This could be due to the fact that dataset 1 was generated such that there are at least 2 neighboring sensors at proximity within 30 meters diameter.

Analysis of the achieved results

The top k% sensors filtering method showed better results than the quality above threshold filtering. Experiments on dataset 1 showed better results than those on dataset 2. The main difference is that for dataset 1, there are neighboring sensors taking measurements at the same time. Generally, giving the sensor completeness the major role in the weights distribution shows better results meaning that sensor completeness is a good indicator of the quality of a sensor. IPI index showed that it either does not have a huge impact on the technique or that it induces the least improvement percentages. This means that the initial evaluation of the sensing units does not necessarily mean that the sensors are going to pursue the same performances in terms of quality.

3.7.5 . Evaluation of extended KNNImpute

We have evaluated our extension of KNNImpute on the two datasets discussed earlier in subsection 3.7.1. The first dataset takes a subset of data from a designated participant where there are data from other participants at a close spatial distance. The second dataset takes the entire dataset of the designated participant.

Table 3.12: Results of KNNImpute for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	56%	20%	24%	6.5
$[100, 0, 0]$	31%	26%	43%	6.7
$[0, 100, 0]$	69%	11%	20%	5.4
$[0, 0, 100]$	13%	69%	18%	7.5
$[60, 20, 20]$	55%	17%	31%	6.6
$[20, 60, 20]$	67%	12%	21%	6.1
$[20, 20, 60]$	44%	36%	20%	6.8
$[40, 40, 20]$	65%	11%	24%	6.4
$[40, 20, 40]$	42%	35%	23%	6.7
$[20, 40, 40]$	60%	20%	20%	6.4

Results on dataset 1

Table 3.12 shows the results of the extension of KNNImpute on dataset 1. The weight configuration $[0, 100, 0]$, that takes only the sensor completeness into account, shows the best improvement result with 69% improved values, only 11% worsened with RMSE 5.4, and 20% unchanged measurements. This shows that giving sensor completeness a bigger weight amplifies the improvement percentage. The second best performing weight configuration is $[20, 60, 20]$, showing 67% improvement of the measurements, 12% worsened and 21% unchanged. Once again, giving sensor completeness the greatest weight results in higher improvement.

The weight configuration $[0, 0, 100]$ shows the worst results with 69% of measurements worsened. This is confirmed given the second worst performing weight configuration is $[20, 20, 60]$. The worsening percentage is significantly smaller because other factors were included, so it was not the only contributing factor. The weight configuration $[100, 0, 0]$ shows the highest value of unchanged measurements: 43%. This configuration considers only the IPI index into account. This can indicate that the IPI index has the least impact on the imputation using the technique.

Results on dataset 2

Table 3.13 shows the results of the evaluation of the proposed extension of KNNImpute on dataset 2. The improved measurements percentage is relatively low. The highest achieved improvement was 35% for both weight configurations $[33, 33, 34]$ and $[20, 40, 40]$, where the weights are distributed among the three quality factors either equally or with slight differences. However, the weight configuration $[20, 40, 40]$ is the most significant because it generated the same improved value as that by the weight configuration $[33, 33, 34]$, but also the lowest worsened value

Table 3.13: Results of KNNImpute for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	35%	12%	53%	61.8
$[100, 0, 0]$	27%	13%	60%	58.7
$[0, 100, 0]$	29%	12%	59%	52.9
$[0, 0, 100]$	31%	20%	49%	72.3
$[60, 20, 20]$	34%	13%	53%	60.3
$[20, 60, 20]$	32%	8%	60%	59.1
$[20, 20, 60]$	33%	11%	56%	66.1
$[40, 40, 20]$	34%	6%	60%	59.7
$[40, 20, 40]$	32%	7%	61%	63.1
$[20, 40, 40]$	35%	6%	59%	62.9

with only 6% worsened measurements. It also shows 59% which is a very high percentage possibly indicating that the sensor quality extension did not have a huge impact on this imputation technique. The unchanged measurements percentages score the highest on average. The maximum value achieved for unchanged measurements is 61% for the weight configuration set $[40, 20, 40]$.

Analysis of the achieved results

The improvement results were disparate among the two datasets studied. The improved measurements reaches significantly higher values with dataset 1 than dataset 2. The best improvement was 69% achieved with the weight configuration [0,100,0] for dataset 1. There are far more measurements unchanged with dataset 2 than for dataset 1, this could mean that when there are neighboring sensors around, less measurements remain unchanged. However, the range of values of measurements worsened were higher for dataset 1 than for dataset 2. This means that the extensions are less prone to errors when the neighboring sensors were not close and disparate. This could be due to the fact that sometimes, the neighboring sensors acquire faulty measurements for some reasons and hence, the imputation technique is more affected by these faulty measurements than other ones.

KNNImpute is fundamentally based on neighboring sensors at proximity, we notice that when the sensor with the missing value that we are trying to impute is at a close distance to other sensors, our extensions perform better.

3.7.6 . Discussion

Generally, giving the sensor completeness the major role in the weights distribution and specifically most of the times the $[0, 100, 0]$ weight configuration shows best results meaning sensor completeness is a good indicator of the quality of a sensor. IPI index showed that it either does not have a huge impact on the tech-

niques in most of the cases, or that it induces the least improved values. This means that the initial evaluation of the sensing units does not necessarily mean that the sensors are going to pursue the same performances in terms of quality. Experiments on dataset 1 showed better results than dataset 2. The RMSE is higher in dataset 2 than in dataset 1. This might be due to the fact that there is a higher possibility for errors in 7 700 measurements than in 480 measurements. For SVDImpute, the top k% sensors filtering method showed better results than the quality above threshold filtering. As for KNNImpute, the improvement results were disparate among the two datasets studied.

3.8 . Conclusion

In this chapter, we presented our proposals on data completeness assessment and improvement. We first presented a multidimensional data model representing any physical element collected using a sensor within the context of a mobile crowdsensing environment. With this model, we have tried to identify the most significant dimensions in order to analyze data in a mobile crowdsensing environment. This model can be used to support different types of analysis such as descriptive analysis using aggregation and mining techniques. Inspired by the multidimensional model, we defined three factors of data completeness: sensor, spatial, and temporal completeness which are applicable to the MCS context. The sensor completeness factor characterizes the completeness of the data coming from sensors, and temporal and spatial completeness study the extent to which the data coming from these sensors are distributed over specified time period and designated spacial area. Then, we proposed metrics to evaluate each of the different proposed factors of data completeness.

We have also addressed completeness improvement and proposed an approach for improving the data completeness for mobile crowdsensing environments. Three data imputation techniques were extended with the quality of the sensor to improve the generation of missing values in the dataset. Our extensions consider that enriching the techniques with the quality of the data measurements improves the quality of their imputation. The extensions with quality prioritize data measurements with higher qualities than others. The approach is evaluated on real data coming from mobile sensors in the opportunistic context of the Polluscope project.

The experiments were done on different weight configurations but proved that giving the sensor completeness factor the major role in the weights distribution of the sensor quality, and specifically, the $[0, 100, 0]$ weight configuration shows the best improvement impact on the results of the data imputation techniques with 71% improvement in the imputation compared to the basic approaches. This

means that our proposed sensor completeness facet is a very good indicator of the quality of a sensor. IPI index showed that it either does not have a huge impact on the techniques in most of the cases or that it induces the least improvement percentages.

Our approach does not tackle the generation of huge missing data chunks in the dataset which could be useful to explore in our future studies. Besides, as the experiments show that the sensor completeness is a very good indicator of the quality of the sensor among what was available for our experiments, it would be interesting to investigate in future studies, the impact of other quality dimensions of the sensor on the imputation. Our approach is limited to three imputation algorithms while it would be interesting to investigate more and different imputation algorithms that are able to impute an entire missing series for example. We only focused in our definition of data completeness on three dimensions of the data, we could in future works refine these definitions and propose new definitions and metrics for the other dimensions of the data that were not addressed in our work.

4 - Anomaly Detection in Mobile Crowdsensing Environments

4.1 . Introduction

One of the most common problems in sensor data is the presence of anomalies. Anomalies can impact and deteriorate the quality of the data. Anomalies are data points or sequences that do not conform to the normal behavior observed. They could manifest as spikes, unusual points, or unusual patterns that often reveal interesting information in the data. There are many causes for the presence of anomalies. Often, the acquisition, integration, and transformation processes can result in faults or errors in the data, such as sensor calibration issues. Among these flaws is the introduction of anomalies. We could also consider anomalies in the data as unusual legitimate data patterns resulting from a normal process but unseen before, which, if not studied with the right perspective or in a special context, could seem faulty and redundant.

Undetected anomalies in a dataset can lead to inaccurate analytics and indicators. For example, the presence of spikes and noise data points can deteriorate the quality of the indicators or the analyses generated from the data. Analytical indicators based on data with undetected anomalies can lead to insights that are a misrepresentation of reality. A sequence of data points (a pattern) that have unusual values could be falsely interpreted as noise when in fact the sequence could indicate interesting information about the data. Such patterns could reveal a fraudulent transaction in transactional contexts, a malicious packet in networking, the malfunctioning of an industrial facility in an air pollution context, or cancer in a medical context. Therefore, detecting anomalies is an essential step in the analysis process because it reinforces the reliability of the predictions and insights and helps better understand the data.

Many anomaly detection techniques are trained on datasets clean of anomalies in order to learn the normal patterns and behaviors in the dataset. The techniques are later tested on datasets with anomalies to detect those that do not conform to the normal learned behaviors. Some anomalies, such as spikes, are very obvious to detect, while others are not as straightforward. Some works are able to detect point anomalies [Bakar et al., 2006], [Zhang et al., 2020]; however, sequences of legitimate point anomalies that reveal unusual behavior remain challenging to detect [Braei and Wagner, 2020]. In many cases, these sequences of point anomalies in the data can only be detected once considered in their specific context [Braei and Wagner, 2020]. Many recent existing works consider the context in the

detection process, such as the work presented in [Zheng et al., 2017]. However, data coming from low-cost nomadic sensors can sometimes be of poor quality due to the erroneous nature of these sensor devices. Low-cost nomadic sensors are prone to points of failure, calibration issues, battery problems, loss of data, and other quality problems. These problems can create a gap between the quality of some data measurements and others. Sensors faced with a quality problem during a time period will produce data measurements of poor quality, while other sensors functioning in perfect conditions can produce high-quality data measurements. To the best of our knowledge, none of the existing works for detecting anomalies in sensor data consider the quality of the measuring sensor. Considering sensor quality and contextual information could improve the quality of the detection of anomalies because they help group patterns with similar values, quality, and context together. A major challenge could be to monitor the quality of the sensor and quantify its quality.

This chapter addresses the detection of anomalies in a mobile crowdsensing environment. It presents a quality-aware approach that aims to detect pattern anomalies using the quality of the sensor and some information about the context. A pattern is a succession of consecutive data points that form a sequence. Our approach aims to show that integrating the quality of the measuring sensors and other contextual information helps improve the quality of anomaly detection. We have explored approaches that aim to improve the detection of anomalies and identified the relevant types of anomalies for the MCS context. We present our approach that integrates the quality of the sensors and the context in which the measurements have been taken, in the anomaly detection process. We first use this information to better group similar patterns together and assign anomaly scores in order to improve the detection of anomalies eventually. We also use it to assign an anomaly score to patterns based on quality such that those that are far away from clusters containing high-quality observations are assigned a higher anomaly score and are more likely to be anomalous. We also evaluate our approach compared to an existing approach which does not take quality into account in order to show the improvement.

This chapter is structured as follows. In section 4.3, we present a general overview of the approach and the intuition behind it. Section 4.2 defines the types of anomalies in mobile crowdsensing environments. In section 4.4 we present an overview of our proposal. Section 4.5 presents our approach on quality-aware anomaly detection for time series. Then, in section 4.6 we present our experiments and evaluation of our quality-based anomaly detection approach on real data. Finally, we conclude the chapter in section 4.7 and present some future works.

4.2 . Types of anomalies in MCS Environments

In this section, we identify the types of anomalies that could be observed in mobile crowdsensing environments and then define each type based on the literature that has long been studying and investigating anomalies in various contexts. We also give examples of each type in different contexts to ease their comprehension.
There are many definitions and types of anomalies discussed in the literature [Hawkins, 1980, Grubbs, 1950, Grubbs, 1969, Braei and Wagner, 2020, Blázquez-García et al., 2021, Fox, 1972, Tsay, 1988]. These are discussed in further detail in subsection 2.5.1. However, to this day, there are no established definitions and types of anomalies across the different application domains and contexts [Carreño et al., 2019].

Anomalies are aberrant data points with values divergent from the norm, be it due to external factors or internal anomalous behaviors. Based on many definitions from the literature, we identify the types of anomalies that are applicable to mobile crowdsensing environments. The mobile crowdsensing environment refers to a context where a large number of mobile devices, equipped with various sensors and connected to the internet, collaborate to collect and share valuable data. Detecting single-point spike anomalies is more straightforward than detecting pattern anomalies [Gupta et al., 2014]. This is because spikes are more obvious to detect by either having a value that is out of the accepted range of the physical element being measured, such as a negative air pollutant level, or by being the only point with a huge sudden drop or rise in value. We identify three types of anomalies in mobile crowdsensing environments: $noise\ anomalies$, $point\ anomalies$, and $pattern\ anomalies$. These anomalies will be detailed in the following subsections.

The context of our work is mobile crowdsensing environments, where mobile low-cost sensors are employed to measure a physical phenomenon in a given crowdsensing context. The collected measurements form a dataset of time series X_n composed of n measurements $\mathsf{X}_n = \{v_{i1}, \ldots, v_{in}\}$ taken by a sensor s_i at consecutive timestamps t_j where $j \in \{1, 2, ..., n\}$.

4.2.1 . Noise Anomalies

Noise anomalies are observations that are introduced to the data through some form of erroneous behavior. In sensor data, a faulty behavior of a sensor unit or a sensor that needs calibration are examples of behaviors that can generate noise anomalies. In the literature, such as in the work of [Braei and Wagner, 2020], a noise anomaly is defined as follows:

Definition 4.2.1 (Noise Anomaly)**.** A noise anomaly is an individual point v_{ij} that deviates significantly from the rest of the data or is not within the normal range.

Noise anomaly points are usually erroneous data points that either have their

values outside the accepted range or are duplicates of other existing data points. For example, a negative value for a physical element that cannot have a negative value or an out-of-the-accepted range value are also examples of noise anomalies. We present an example hereafter.

Example 4.2.1. In the context of air quality sensing, a data point indicating $PM_{2.5}$ in the air cannot have a negative value of -999,895,666 $\mu g/m^3$ or -3,455 $\mu g/m^3$. Such observations are considered noise points because it is impossible to have a negative value of air pollutant, and it is not realistic to have a measurement of 9,895,666 $\mu g/m^3$ in a series where other time-adjacent measurements have values $\leq$ 50 $\mu g/m^3$.

Noise anomalies are of no significant value for data analysts as they distort the analysis [Aggarwal, 2016]. Cleaning the dataset of these noise points is necessary as a pre-processing step before analyzing the data.

4.2.2 . Point Anomalies

According to the works of [Blázquez-García et al., 2021], [Braei and Wagner, 2020], point anomalies in the data are data points that are different from the other adjacent data points, and are defined in these works as follows:

Definition 4.2.2 (Point Anomaly)**.** A point anomaly is a single faulty measurement v_{ij}, whose value is remarkably different from the dominant majority distribution of the data and, more specifically, different from its neighboring measurements, spatially or temporally.

However, unlike the noise anomalies, a point anomaly, despite having aberrant values compared to neighbors, has a value within the accepted range of the studied pollutant. In time series taken by some sensor s_i at timestamp t_j, a point v_{ij} is a point anomaly if its value is within the accepted range but also that is either too high or too low compared to its neighboring measurements $\{v_{ij-k}, v_{ij-k+1}, \ldots, v_{ij}, ..., v_{ij+k}\}$, k defines the length of the range of accepted values. Considering the spatial aspect into account also indicates that a data point v_{ij} is recognized as an anomaly if it has a value that is significantly different from other measurements that are at a distance d to v_{ij} that is less than or equal to a certain threshold δ. We illustrate this in the following example 4.2.2.

Example 4.2.2. Consider, for example, the context of air quality monitoring collecting a time series of the pollutant NO_2 with the following measurements X_n = {10, 11, 9, 12, 892, 10, 9, 11, 8}. We assume k=4 and δ = 450, where the data point having a difference with all of its neighboring measurements greater than this threshold is considered anomalous, the measurement with 892 value is a point anomaly given the difference between the value of the measurement and all of its eight neighboring measurements is > 450.

4.2.3 . Pattern Anomalies

Several works have defined pattern anomalies (subsequence outlier) [Blázquez-García et al., 2021], [Braei and Wagner, 2020] as a sequence of data points whose joint behavior is unusual. In these works, a pattern anomaly is defined as follows:

Definition 4.2.3 (Pattern Anomaly)**.** A pattern anomaly is a sequence of consecutive data measurements $\{v_{ij-k}, ..., v_{ij}, ..., v_{ij+k}\}$, where the value of each measurement v_{ij} falls within the accepted limits of the distribution of the dataset yet, the sequence of these measurements shows an anomalous pattern or behavior.

Pattern anomalies are defined as a sequence of measurements that do not conform to the norm compared with most sequences in the dataset. A pattern anomaly could be a sequence of normal data points that form a pattern known to be anomalous.

Example 4.2.3. Consider a series of data measurements representing temperature values measured in degrees Celsius TM= {35°, 1°, 35°, 1°, 35°, 1°}, with measurements taken every minute on a wind-free quiet night, because it does not make sense that temperature value changes from 80 to 1 in one minute.

A pattern anomaly could also be a sequence of data points that all have aberrant values outside the observed values in the dataset. These anomalous patterns can be of a high value if recognized, such as heatwaves, nearby fuel combustion, and a tornado in air quality data.

Example 4.2.4. A pattern anomaly where a series of particulate matter $PM_{2.5}$ measurements $X_n = \{v_{ij-3}, ..., v_{ij}, ..., v_{ij+3}\}$ acquired by a sensor s_i with values {88, 87, 85, 89, 86, 84, 88} $\mu g/m3$ taken after midnight at timestamps $\{t_1, \ldots, t_7\}$ respectively. These $PM_{2.5}$ values are within the range of what can sometimes be observed for this pollutant, but, at this time and location, this succession of values is not usually recorded. Hence, this pattern could be anomalous.

The example above shows that when the measurement values do not align with the surrounding context, it could be due to an interference in the sensor unit between specific air pollutants, causing the sensor to read aberrant values, or sometimes due to problems in the data integration or loading processes or even at times can be explained with additional contextual information that would explain the behavior observed.
In this chapter, we are going to address the detection of pattern anomalies. We will later introduce our approach that detects pattern anomalies for data in a mobile crowdsensing environment.

4.3 . General approach for anomaly detection

This section describes the general overview of our proposed anomaly detection approach which aims to detect pattern anomalies in time series data from sensors. We propose a quality-aware anomaly detection approach based on grouping

subsequences of time series into clusters comprising subsequences of similar behaviors taking quality into account and the context information. The idea of our proposal is to show that considering the quality and the context during the detection process, essentially the quality of the measuring sensors, improves the detection of anomalies. We also want to show that adding contextual information to the processing offers more insightful similarity computation between the data sequences, eventually improving the detection results.

Figure 4.1 illustrates the general workflow of our anomaly detection approach. The approach is composed of three components, which are described as follows. The data transformation component is responsible for transforming the time series input into a set of subsequences of a fixed length. This transformation allows us to study the patterns in the time series instead of single data observations. Given inputs of several time series $X_n = \{v_{i1}, \ldots, v_{in}\}$ taken by sensor s_i, a window size *w* that determines the number of data points that constitute one subsequence/window, and a sliding step length L that determines the increment step. During the data transformation step, the input time series X_N is transformed into a set of subsequences S where each subsequence has a fixed length *w*. The output of this component is a set of subsequences, each subsequence containing *w* data points such that the clustering is done on patterns instead of single data points in order to identify pattern anomalies.

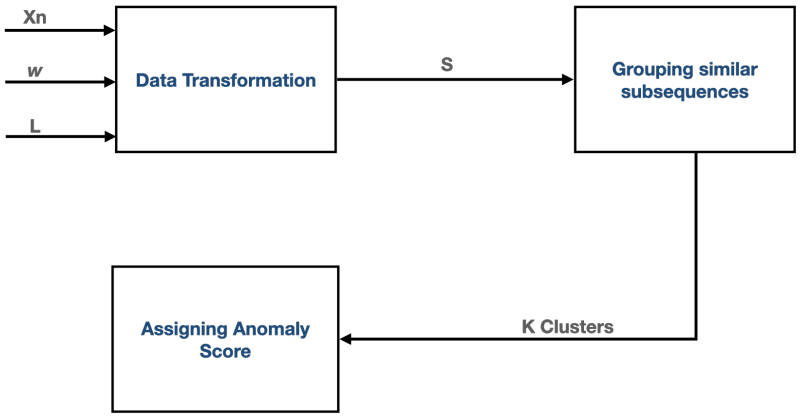

Figure 4.1: A workflow showing the steps of our proposal.

The data grouping component handles the clustering of the subsequences into k-clusters of similar subsequences taking quality and context into account. It takes the subsequences as input from the data transformation component

and groups them into k clusters while considering the quality of the measuring sensors and the context information. This component groups subsequences that exhibit similar patterns or characteristics together. The goal is to identify clusters of subsequences that share common features and behaviors such that normal subsequences are grouped in some clusters, and anomalous subsequences are grouped together in other clusters. The output of this component is the set of k clusters with k centroids comprising the entire set of subsequences in each cluster.

The last component is the assignment of an anomaly score. This component leverages quality to assign anomaly scores to all the subsequences in the dataset. In this component, we assess the quality of a cluster and then propose a scoring method that assigns an anomaly score value to each subsequence depending on its distance from clusters with a certain quality index. An anomaly score e_i is computed for every subsequence s_i where $i \in \{1, \dots n\}$, n is the number of data points in initial series X_n to form a set e comprising all the anomaly scores of the subsequences in S. This set of computed anomaly scores helps identify anomalous subsequences from the normal ones. All subsequences with an anomaly score above a certain predefined threshold are considered anomalous.

4.4 . Using k-means for anomaly detection

We provide in this section some preliminaries about the existing anomaly detection method we founded our quality approach on. We present the existing approach with the techniques it uses. We present the k-means algorithm and the approach that uses it for anomaly detection. Numerous machine learning approaches use a variety of techniques for anomaly detection in time series. Recently, many works in anomaly detection have focused on time series data. Many approaches have worked on contextual anomaly detection [Tsay et al., 2000],[Liang and Parthasarathy, 2016],[Zheng et al., 2017]. Contextual anomaly detection approaches leverage information about the context to better detect anomalies in the data. There are five categories of anomaly detection approaches: *statistical-based* [Braei and Wagner, 2020], *distance-based* [Tran et al., 2016], *clustering-based* [Breunig et al., 2000a], *deviation-based* [Giannoni et al., 2018], and *neural network-based* [Audibert et al., 2020]. The details of each category of approaches are discussed in more detail in section 2.5

This section describes the subsequence clustering approach for anomaly detection based on the k-means clustering algorithm, an unsupervised learning technique. This approach is a clustering-based anomaly detection approach. This family of works groups similar data together and can detect pattern anomalies. k-means is a common clustering technique tested on time series for anomaly

detection [Braei and Wagner, 2020]. We first describe the method and present its principles. Then, we discuss in detail how k-means is used in the work of [Braei and Wagner, 2020] to detect anomalies in time series.

4.4.1 . The k-means algorithm

k-means is an unsupervised machine learning algorithm used to cluster data observations into k clusters depending on their similarities [MacQueen, 1967]. The k-means algorithm minimizes the intra-cluster distances and maximizes the inter-cluster distances. k-means uses the expectation maximization approach to solve the problem of assigning data observations to the closest clusters and computing new centroids [Bishop, 2016]. Assume we have a set containing N data observations x_1,x_2,...,x_N where each x_i is D-dimensional. The objective function minimizes the distance between the data points and the cluster centroids while assigning them to clusters so as to maximize the similarity between the subsequences within the same cluster. It represents the sum of the squares of the distance of each data point to its assigned centroid μ_p multiplied by w_{ip} which is a binary indicator variable that will be assigned a value of 1 if data observation x_i belongs to cluster p, and a value of 0 otherwise. We first randomly choose initial values for $\mu_p, \forall$ *p* $\in$ {1,2..,k}. According to [Bishop, 2016], the objective function θ is computed as shown below:

$$\theta = \sum_{i=1}^{N} \sum_{p=1}^{\mathrm{k}} w_{ip} \|x_i - \mu_p\|^2$$

Where μ_p is the centroid of cluster p, N is the number of data observations in the dataset.
The goal is to find values for the w_{ip} and the values of μ_p so as to minimize θ. Hence, it is a minimization problem of two parts: minimizing by differentiating θ w.r.t w_{ip}, then w.r.t μ_p. We first assign each data observation x_i to the closest cluster based on its distance from the cluster centroids as shown in the formulas below. This results in w_{ip} having a value of 0 if data observation x_i does not belong to cluster p, otherwise, it will be assigned a value of 1 when the distance between this data observation and centroid μ_p of cluster p is the minimum among other centroids.

$$\frac{\partial \theta}{\partial w_{ip}} = \sum_{i=1}^{|S|} \sum_{p=1}^{\mathrm{k}} \|x_i - \mu_p\|^2$$

$$\Rightarrow w_{ip} = \begin{cases} 1 & if\ p = argmin_j \|x_i - \mu_j\|^2 \\ 0 & otherwise \end{cases}$$

The derivative is then computed w.r.t μ_p and the centroids of each cluster are later recomputed to reflect the new assignments of the subsequences in the clusters:

$$\frac{\partial\theta}{\partial\mu_p} = 2\sum_{i=1}^{|S|} w_{ip}(x_i - \mu_p) = 0 \Rightarrow \mu_p = \frac{\sum_{i=1}^{|S|} w_{ip}x_i}{\sum_{i=1}^{|S|} w_{ip}}$$

The difference achieved by the objective function θ is reduced with each iteration until the algorithm converges at some point. However, it may converge to a local minimum rather than the global minimum of θ [MacQueen, 1967]. The two steps of assigning data observations to clusters and then recomputing the centroids of the clusters are repeated until the algorithm converges or until it reaches the maximum number of iterations set. When the algorithm converges, there are no new assignments of subsequences to clusters, and the computed cluster centroids are the same as with the previous iteration.

4.4.2 . Using k-means for subsequence time series anomaly detection (STSC)

Clustering techniques such as k-means [MacQueen, 1967] have been used for anomaly detection tasks in several application domains. One of the existing approaches for anomaly detection in time series using k-means is the work presented in [Braei and Wagner, 2020]. The authors in this work employ a sliding window-based approach where the algorithm clusters subsequences of the dataset instead of single data observations.
Given a time series X_N with N data observations over N timestamps: $\{X_N\} = \{x_1, x_2, ..., x_N\}$, a window length w, and a sliding step length γ, the time series $\{X_N\}$ results in a set of subsequences S $\subseteq \mathbb{R}^{(N-w)\times w}$:

$$S = \{(x_0, x_1, ..., x_w)^T, (x_{0+\gamma}, x_{1+\gamma}, ..., x_{w+\gamma})^T, ..., (x_{N-w}, x_{N-w+1}, ..., x_N)^T\}$$

k-means clustering is applied on the data after a preprocessing step that transforms the series of data measurements to a set of subsequences S, where |S| is the magnitude of set S, signifying the number of subsequences in S. The input data for k-means in this approach is the set S = $\{ s_1, ..., s_{|S|} \}$ which is constituted of subsequences of data observations instead of regular data points.

The approach uses the Euclidean distance as the similarity metric to compute the assignments of subsequences to clusters and to compute the distances between subsequences and centroids. k-means is later applied on the set of subsequences S until it converges or reaches the maximum number of iterations limit, resulting in k clusters with their corresponding k centroids. The centroids are computed as the mean of the subsequences in the cluster they belong to.
For each subsequence in S, a value is computed as the anomaly score corresponding to each subsequence. This anomaly score is calculated as the Euclidean distance

of each subsequence in S to its nearest centroid.

$$e_i = min(d(s_i - c)) \, \forall \, c \, \in C$$

Where C is the set of the centroids computed by the algorithm, s_i is subsequence at index i , and d is the Euclidean distance.
The set ε comprises the computed anomaly scores of all the subsequences in S as follows:

$$\varepsilon = (e_0, e_1, ..., e_{|S|}) \quad where \, e_i \, for \, i \in \{0, ..., |S|\}$$

Finally, to distinguish anomalous subsequences from normal ones, a threshold is defined to identify all subsequences with an anomaly score above this threshold, as anomalous subsequences. A subsequence $s_i \in S$ is an anomalous subsequence according to the following:

$$\begin{cases} e_i > \delta & s_i \text{ is anomalous} \\ e_i \leq \delta & s_i \text{ is not anomalous} \end{cases}$$

Where $\delta \, \in \mathbb{R}$ is a predefined threshold. The approach is described by algorithm 1 bellow.

4.5 . Quality-aware anomaly detection approach

In this section, we present our quality-based anomaly detection approach that uses sensor quality and contextual information to improve the detection of anomalies. The idea behind our approach is to primarily introduce the quality of the sensors into the anomaly detection process. The quality of a data point x_{ij} is represented by the quality of the sensor s_i capturing this data point over the time period T, including the time when the data point x_{ij} was taken.

We also introduce context information that makes the clustering algorithm aware of the context surrounding the measurement at the time it was taken. Contextual information is used in order to improve grouping similar subsequences together.

Our goal is to show that introducing quality and context information in the clustering and assigning anomaly scores can improve the detection of anomalies because understanding the context implies a better understanding of the data.

In this section, we first present the quality of the sensor in subsection 4.5.1 and the context information in subsection 4.5.2 that are used to improve existing anomaly detection techniques. We then describe the steps of our approach within each component according to the general architecture presented in Figure 4.1. We discuss how we improve anomaly detection by introducing information about the context and the quality of each measurement to the anomaly detection process.

Algorithm 1 STSC for anomaly detection

Input: Time series X_N, window size w, sliding step length l, k number of clusters, anomaly threshold δ.
$S \leftarrow \{(x_0, x_1, ..., x_N)^T, (x_{0+\gamma}, x_{1+\gamma}, ..., x_{N+\gamma})^T, ..., (x_{N-W}, x_{N-W+1}, ..., x_N)^T$
$\mu_p \leftarrow$ random subsequence from $S \quad \forall\ p \in \{1, 2.., \mathrm{k}\}$
for $s_i \in S$ **do**
    $d \leftarrow argmin_j \|s_i - \mu_j\|^2 \quad \forall\ j \in \{1, 2.., \mathrm{k}\}$
    if $s_i \in$ cluster c_p **then**
        $w_{ip} \leftarrow 1$
    else
        $w_{ip} \leftarrow 0$
    end if
end for
for p $\in$ {1,2..,k} **do**
    $\mu_p \leftarrow \frac{\sum_{i=1}^{|S|} w_{ip} s_i}{\sum_{i=1}^{|S|} w_{ip}}$
end for
Repeat steps 4-14 until convergence
for $s_i \in S$ **do**
    $e_i \leftarrow min(d(s_i - c_j))\ \forall\ j \in \{1, 2.., \mathrm{k}\}$
    if $e_i > \delta$ **then**
        $s_i \leftarrow$ anomalous
    else
        $s_i \leftarrow$ not anomalous
    end if
end for

4.5.1 . Sensor quality for anomaly detection

This subsection presents the quality of the sensor that we will introduce into our anomaly detection approach. The quality of the sensor is used in the clustering and anomaly scoring steps.

The quality of the sensor gives information about the quality of the sensing unit taking the measurement. In order to characterize the quality of the measurement, the quality of the device that took the measurement at the time it was captured is considered. We define sensor quality and provide categories of measures to assess the quality of a sensor. We then propose an aggregation that computes a final global value of the quality of a sensor.

The quality of the data measurements is defined by the quality of the measuring sensors. The assessed quality value of a sensor at a certain timestamp is associated with the measurements that are taken around this time. The quality of a data measurement is the quality of the sensor taking this measurement around the time it was taken.

A variety of external and internal circumstances can have a significant impact on the performance of a sensor unit. Some of these circumstances have to do with the device itself, others have to do with external factors, and others have to do with the usage of the sensor. Two sensors of the same type and from the same manufacturer can perform differently after a certain time.

We define three categories of metrics to evaluate the quality of the sensors as follows: metrics at the device level such as initial evaluations of the sensor units, others related to the device usage such as studying accuracy after a sensor has been used for a period of time, and others related to the comparison with reference data such as studying correlations between actual data from the sensor and reference data. The assessed quality factors of the data using different metrics from each category, are finally aggregated to compute a global quality score. In order to aggregate the values resulting from different quality evaluations, all these values must be normalized. Once the values of the assessed factors are normalized, we compute a global quality score that represents the overall quality of a single data measurement.
We recall the three categories of metrics introduced in chapter 3 (in subsection 3.5.1) as follows.

Device-related metrics assess the quality at the device level. It could be related to the physical characteristics of the sensor itself. An example of a device-level metric is the sensor's capacity to capture a physical element in the presence of challenging factors such as heat, wind, etc. *Usage-related* metrics assess the performance of a sensor while being used or after a period of time. It is impacted

by how the sensor is being used by the carrier and the external conditions it was exposed to. For example, the completeness of the measurements collected by a sensor. *Reference-related* metrics compare the data to other reference data. For example, studying the correlation of the actual data with reference data.

In order to compute a global quality score of the sensor at a timestamp t_j, an aggregation method is employed to combine multiple quality values obtained from different categories of quality metrics. The evaluation of a certain quality factor of the sensor at a timestamp t_j is done for all the data measurements recorded by this sensor starting at timestamp t_1 that marks the time of the last evaluation of this factor for this sensor, up until timestamp t_j. This means that the data measurements from this sensor that were recorded after the time of the last registered evaluation will be included in the next evaluation, and all the measurements taken by this sensor until timestamp t_j will be included in this new evaluation.

The values of the assessed quality factors are first normalized. Normalizing the values involves scaling or transforming them to a common range or standard, ensuring that they are comparable and facilitating meaningful aggregation. Our aggregation process enables an evaluation of data quality by considering a single or multiple quality values as follows.

Definition 4.5.1 (Sensor Quality at Timestamp t_j)**.** Sensor quality at a timestamp t_j is a weighted aggregation of the quality values resulting from the assessment of a set of quality factors for this sensor. For each factor, the considered value is the closest in time to timestamp t_j.

$$Q_{s_k,t_j} = \frac{\sum_{i=1}^{n} w_i \times qm_i}{n} \tag{4.1}$$

Where n is the number of considered factors, qm_i is the quality value of factor qf_i, which is the closest in time to t_j, s_k is the assessed sensor, and w_i is the weight assigned to each quality value qm_i. The sum of all weights w_i $i \in \{1,\ldots,n\}$ is equal to 1.

4.5.2 . Contextual information for anomaly detection

Information about the context are related to the environment or the context surrounding the sensor while collecting measurements about a physical element in the real world. This information about the context could be intrinsic, related to the sensing device internally; or extrinsic related to the outside world and the factors surrounding the device. For example, the battery level is intrinsic context information while the temperature, wind speed, and humidity are extrinsic information.

Contextual information can reveal important relationships, dependencies, or patterns that may not be apparent from the data alone. This descriptive context aids in uncovering patterns, relationships, and dependencies that contribute to a deeper analysis and interpretation of the data.

The external conditions surrounding a sensor unit at a certain timestamp can significantly impact the interpretation of the measurements taken by this sensor at that time. In air quality contexts, for instance, several air components, device-intrinsic, and other external factors directly impact the levels of some air pollutants, making them strongly correlated.

In the context of meteorological data, the authors of [qin Han et al., 2011] found that a polynomial relationship exists between O_3 and NO_2/NO. The study presented in [Liu et al., 2020] has shown that the concentration of air pollutants at most measuring stations was significantly negatively correlated with wind speed, precipitation, and relative humidity but positively correlated with atmospheric pressure. For instance, contextual information could be the user's activity in the context of air quality monitoring that can justify the high pollutant levels. Therefore, factors in the surrounding context can noticeably impact the understanding of a measured element.

Definition 4.5.2 (The Context)**.** The context is defined as a set of features C_{ij} surrounding a data measurement v_{ij} that helps better understand it. Each contextual feature cf_k has a value cv_k that describes a relevant aspect of the physical surrounding environment of a data measurement to enhance its understanding.

$$CF = \{(cf_k, cv_k)\} \qquad k \in \mathbb{N}^+$$

In the context of air quality monitoring using mobile sensors, the contextual information could indicate the environment where the measurement was taken. This information can be very useful because air pollutant levels differ from one context to another. For example, the air is 10 times more polluted near the metros than in the outdoor streets.

4.5.3 . Data transformation

The process of data transformation is responsible for transforming our input dataset from a set of time series collected by several sensors into one set comprising all the subsequences constructed. It is represented by the data transformation component in Figure 4.1. This step is similar to the transformation of the data discussed in the STSC approach in subsection 4.4.2.

Our dataset is composed of a set of N time series $X_N = \{X_1, \ldots, X_N\}$ each collected by a sensor s_i. Each series collected by a sensor s_i has a set of

n measurements $X_n = \{v_{i1}, \ldots, v_{in}\}$. In a mobile crowdsensing context, each measurement vector v_{ij} is taken by sensor s_i, has a value v, timestamp t_j and other features such as the geographic coordinates (latitude, longitude), and the set of contextual information about the surrounding context CF. Hence, the measurement vector v_{ij} is represented by: $v_{ij} = v_{ij} = (v, s_i, t_j, (lat, lon), s_i)$ which is a general set of some features that represent a measurement in a mobile crowdsensing environment (MCS).

The time series is first transformed into a set of subsequences each having a fixed size created with a sliding step length that is also predefined. The resulting set of subsequences will be the new dataset where each subsequence will be treated as a data point.
Given a time series with N measurements $X_N = \{v_{i1}, ..., v_{iN}\}$, where each has a number of measurements in $\in \mathbb{N}^+$, a window length w, and a sliding step length γ, the time series $\{X_N\}$ results in a set of subsequences S $\subseteq \mathbb{R}^{(N-w)\times w}$:

$$S = \{(v_0, v_1, ..., v_w)^T, (v_{0+\gamma}, v_{1+\gamma}, ..., v_{w+\gamma})^T, ..., (v_{N-w}, v_{N-w+1}, ..., v_N)^T\} \tag{4.2}$$

Example 4.5.1. Suppose we have a set of time series X_N that has the following values $X_N = \{20, 21, 22, 23\}$, a window length 2, and a sliding step 1. The resulting set of subsequences is S = {(20,21), (21,22), (22,23).}

The window length w can be determined empirically in the experiments. The number of sequences P can be computed as follows:

$$|S| = \frac{N}{w} + (w - \gamma)$$

N is the total number of measurements, w is the fixed subsequence length defined earlier, and γ is the sliding step length.

The clustering algorithm is then applied to the set of subsequences instead of single data points. The input data for k-means in this approach is the set S, composed of subsequences of data observations instead of regular data points:

$$S = \{s_1, ..., s_{|S|}\}$$

Where |S| is the number of subsequences in set S, and s_i is a subsequence, i is the index of the subsequences with values $\in \{1, \ldots, |S|\}$.

4.5.4 . Quality-based data clustering

In this component, we use the k-means clustering algorithm to group subsequences of similar behavior together. We enrich the clustering process with information about the quality of the sensor and the context. The quality of the sensor is propagated to the measurements in each subsequence. It is possible to

have data measurements from different sensors within the same subsequence. We introduce our notion of subsequence quality.

Definition 4.5.3 (Subsequence Quality)**.** Subsequence quality (q_{s_k}) is a quality vector that indicates the quality of the data points in a subsequence s_k. It is a subsequence of fixed predefined length w comprising the corresponding quality values of each one of the data observations that constitute the subsequence s_k. Hence, if the subsequence s_k has the measurements $\{v_1, v_2, v_3\}$, it has a subsequence quality $q_{s_k} = (q_{v_1}, q_{v_2}, q_{v_3})$.

Computing the similarity between two subsequences:
To compute the distance between two subsequences, we propose a quality-aware similarity function that uses the Euclidean distance and takes into consideration the quality of the subsequence and the contextual information in addition to the values of the measurements. The subsequence is a vector that contains the measurements, and the quality of the subsequence is also a vector containing the quality of each measurement in the subsequence. Finally, the contextual information is also represented by a vector of values indicating the context of each measurement in the subsequence. The context values are encoded to integers to facilitate the computation of the similarity. The similarity between two context vectors is computed as one if all the values are equal; otherwise, zero. Hence, we compute the distance between two subsequences s_1 and s_2 using the L_2 norm as follows:

$$D_{s_1,s_2} = \|s_1.q_{s_1}.cv_{s_1} - s_2.q_{s_2}.cv_{s_2}\|_2 \tag{4.3}$$

Where s_i is a subsequence of measurements, q_{s_i} is the subsequence quality vector of s_i, and cv_{s_i} is the context vector of subsequence s_i.

Assigning subsequences into clusters

To assign a subsequence s_i to a cluster p, the distance between the subsequence and the centroid of the k clusters is computed. Subsequence s_i is assigned to the closest cluster in the distance. The value of w_{ip} indicates whether a subsequence s_i belongs to cluster p or not:

$$w_{ip} = \begin{cases} 1 & if\ p = argmin_j \|s_i.q_{s_i}.cv_{s_i} - \mu_j.q_j.cv_j\|^2 \\ 0 & otherwise \end{cases} \tag{4.4}$$

Where μ_j is the centroid of cluster j, q_j is the quality vector of the centroid of cluster j, and cv_j is the context vector of the centroid of cluster j.

Computing the cluster centroids

After assigning subsequences into clusters, we need to recompute the centroids of the clusters. For the computation of the centroids, we also take into account the quality of each subsequence in computing the centroids. Where each considered subsequence is weighted by its quality, as shown in the following equation:

$$\mu_p = \frac{\sum_{i=1}^{|S|} w_{ip}.s_i.q_{s_i}.cv_{s_i}}{\sum_{i=1}^{|S|} w_{ip}.q_{s_i}.cv_{s_i}} \tag{4.5}$$

The objective function of the clustering is a quality-aware objective function θ that considers the quality of the measurements. w_{ip} is assigned a value of 1 if a subsequence s_i belongs to cluster p. Otherwise, w_{ip} will have a value of zero, indicating that it does not belong to this cluster. The objective function θ is defined as follows.

$$\theta = \sum_{i=1}^{|S|} \sum_{p=1}^{\mathrm{k}} w_{ip} \| s_i.q_{s_i}.cv_{s_i} - \mu_p \|^2 \tag{4.6}$$

Where q_{s_i} is the quality of subsequence s_i, μ_p is the centroid of cluster p, and cv_{s_i} is the context information vector of subsequence s_i.

4.5.5 . Assigning anomaly score and detecting anomalies

This subsection presents the computation of an anomaly score to all the subsequences in the dataset taking the quality of the sensor into account. It also describes the approach to identify anomalous subsequences from non-anomalous ones.

Computing anomaly score

After the subsequences have been clustered into k-clusters taking quality and context into account, a quality anomaly score for every subsequence in the dataset is computed. We first introduce the concept of high-quality cluster. The anomaly score of a subsequence is then computed as the distance from this subsequence to its nearest high-quality cluster. Hence, if a subsequence belongs to a high-quality cluster, it will be assigned a lower anomaly score than another subsequence that does not belong to a high-quality cluster. This is because we assume that clusters with high quality are not anomalous.

The centroid of a cluster is computed as the average of the subsequences in the cluster. The quality of the centroid is computed as the average of the quality of the subsequences in this cluster. The cluster quality q_{C_k} is a single value that

represents the quality of the whole cluster C_k. It is computed using the quality of the centroid μ_k of cluster C_k. The centroid of the cluster is a subsequence composed of L measurements. Each measurement is characterized by a quality value computed as in definition 4.5.1. The cluster-quality q_{C_k} is defined as follows.

Definition 4.5.4 (Cluster Quality)**.** The quality of a cluster C_k is computed as the aggregation of the quality values corresponding to all the measurements in the centroid subsequence μ_k:

$$q_{C_k} = \frac{\sum_{j=1}^{L} q_{v_j}}{L}$$

Where L is the fixed subsequence length, and q_{v_j} represents the quality of a measurement v_j in μ_k of cluster C_k. q_{v_j} is computed according to definition 4.5.1.

We provide hereafter, an example of cluster quality in example 4.5.2.

Example 4.5.2. Suppose we have a cluster C_1 with a centroid that has the following quality vector q_{μ_1} = {0.6,0.4,0.6,0.8}. Then the quality of cluster c_1 is computed as avg(0.6 + 0.4 + 0.6 + 0.8) = 0.6.

Based on cluster quality, the anomaly score of a subsequence is computed while taking the quality of the clusters into account. The anomaly score is greater for subsequences that do not belong to a high-quality cluster than for subsequences that do belong to a high-quality cluster.

Definition 4.5.5 (High-Quality Cluster)**.** A cluster is said to be of high quality if its cluster quality index is greater or equal to a predefined threshold. This threshold is denoted by α and could be computed empirically in the experiments or predefined by a domain expert.

Assume a dataset of time series that is transformed into a set of subsequences S = $\{s_1, \ldots, s_{|S|}\}$. An anomaly score e_i is computed for each subsequence s_i in S.

Definition 4.5.6 (Anomaly Score)**.** The anomaly score e_i assigned to a subsequence s_i is computed as the multivariate Euclidean distance of subsequence s_i to the nearest high-quality cluster centroid μ_k according to the following :

$$e_i = \begin{cases} d(s_i, \mu_j) & for\ q_{C_j} \geq \alpha \\ min(d(s_i, \mu_p)) \qquad \forall\ \mu_p,\ p \in \{1, ..., n\},\ q_{C_p} \geq \alpha & for\ q_{C_j} < \alpha \end{cases} \tag{4.7}$$

Where s_i is a subsequence $\in C_j$, μ_j centroid of cluster C_j, $\forall\ j \in \{1, ..., n\}$, q_{C_j} is quality of cluster C_j, and α is a predefined threshold.

This means that if a subsequence already belongs to a high-quality cluster C_j, the distance would be between the subsequence and the centroid μ_j of the cluster C_j. Otherwise, if a subsequence belongs to a cluster with a quality that falls below the predefined threshold, the anomaly score would be the distance between the

subsequence and the nearest centroid of a high-quality cluster. This is illustrated in example 4.5.3.

Example 4.5.3. Figure 4.2 shows a subsequence s_1 belonging to cluster C_1. We set the threshold $\delta = 0.7$. Hence, cluster C_1 is a high-quality cluster because it has a quality $q_{C_1} = 0.80$. The anomaly score e_1 is the distance between s_1 and the centroid μ_1 of cluster C_1 since the subsequence belongs to a high-quality cluster. Subsequence s_2 belongs to cluster C_2 of quality $q_{C_2} = 0.20$. Cluster C_3 also is a high-quality cluster with quality $q_{C_3} = 0.75$, the anomaly score e_2 of s_2 is the distance between subsequence s_2 and the nearest centroid of a high-quality cluster, in this case, μ_1 because centroid μ_1 of cluster C_1 is closer to s_2 than centroid μ_3.

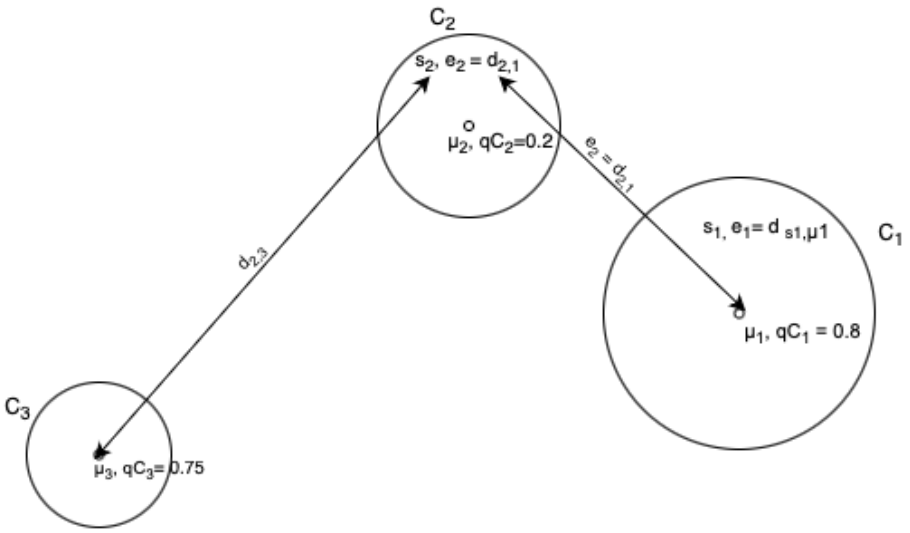

Figure 4.2: Computing anomaly score of a subsequence that belongs to low-quality cluster.

Identifying Anomalous Subsequences

Subsequences that do not belong to a high-quality cluster have a higher possibility of being anomalous. Hence, once the anomaly score of all subsequences in the dataset is computed, we diagnose a subsequence as anomalous if its corresponding anomaly score is greater than a predefined threshold δ. Any subsequence with an anomaly score less than the threshold δ is considered non-anomalous.

Definition 4.5.7 (Anomalous subsequence)**.** An anomalous subsequence is a subsequence whose anomaly score is greater than a predefined threshold δ.

4.6 . Evaluation of Quality-Aware Anomaly Detection Approach

This section introduces the experiments and evaluations of the proposed quality anomaly detection approach on time series data collected within a mobile crowdsensing environment. Our experiments aim to show whether introducing quality and context information into anomaly detection would improve the detection of anomalies. We evaluate our quality-aware anomaly detection approach compared to an existing approach that uses the k-means algorithm for subsequence time series clustering-based anomaly detection [Idé, 2006].

4.6.1 . Context and Datasets

The evaluations are done on air quality data collected using nomadic low-cost sensors within the context of an opportunistic air quality project, Polluscope [Brahem et al., 2021]. Opportunistic crowdsensing means that sensor carriers go by their daily routines without changing their routes or usual destinations. The observed pollutant is measured by low-cost nomadic sensor units along with timestamps and geographic location. The collected data is a set of time series. The data acquisition process within the project was divided into three campaigns of different durations. During each campaign, volunteer users carried a set of sensors with them over their daily routines. The carried sensors collected data measurements of a different pollutant annotated with time, geolocation, and context information. In our settings, we consider the context as the activity that represents the environment that the sensor carrier is in.

The experiments are conducted on time series measuring the particulate matter air pollutant of diameter 2.5 $PM_{2.5}$. The data used in the experiments were acquired between 1^{st} of January 2020 and the end of March 2020 by several users and several sensors measuring the same pollutant.

The data were manually annotated by the sensor carriers to describe the context of the location they were in at different times of their day, such as home, work, shops, restaurants, cars, metro, etc. This gives us insights into the context where the measurements were taken. In our experiments, the context can either be indoors or outdoors. We have selected the subset of the initial dataset for the experiments where the context of data measurements was manually validated to ensure the annotations are correct.

We have performed two types of experiments described as follows.

- Experiment type 1: in this experiment, we compare our approach to the baseline approach while varying the window length w, referred to as the subsequence length, number of clusters k, anomaly threshold δ, and percentage of injected anomalies. The dataset of this experiment contains measurements of the $PM2.5$ pollutant. It comprises 33 000 data measurements and has been selected for which the annotations were confirmed. This

dataset is referred to in the sequel as dataset 1.

- Experiment type 2: in this experiment, the parameters including the window length, number of clusters, and anomaly threshold, were concluded from the type 1 experiment. Hence, for this experiment, these parameters were fixed, and we only varied the percentage of injected anomalies. In this experiment, we refined dataset 1 by further cleaning 20 000 data measurements from dataset 1. We will refer to this dataset as dataset 2.

4.6.2 . Methodology

The experiments are done on an Apple M1 chip processor with 16GB RAM. We used Python 3.9.7 on Jupyter Notebook to automate the pipeline of the experiments as well as to generate anomalies using our customized anomaly generator subsection 4.6.3. We conduct two types of experiments.

We perform the first type of experiments on dataset 1 while varying some parameters of the approach. The experiments include various percentages of injected anomalies: 1%, 2%, 3%, and 4%, we also vary the window length w also referred to as the subsequence length, the number of clusters k, the anomaly threshold δ. We aim to compare the achieved results from the baseline with the results of our approach. Another goal of this type of experiment is to help us learn the best parameter values of the anomaly detection approach.

We later conduct the second type of experiments on dataset 2 but with increasing data anomaly percentages injected 5%, 10%, 15%, 20%, and 25% since the type 1 experiments showed that the approaches could work better with higher percentage of injected anomalies. We set the values of the following parameters of the approach: the window length $w = 7$, the number of clusters k=4, and the anomaly threshold $\delta = 0.7$, both on the baseline and our approach. We have repeated this experiment three times, each time with a new random generation and injection of the various percentages of anomalies tested.

Before all the experiments, the initially selected dataset for the experiments was not clean of anomalies. Hence, we manually clean the data from spikes. We also remove data measurements with a negative value because it is physically impossible to have a negative value of the measured element we studied in our experiments.

To test the usefulness of our approach, we have implemented an anomaly generator that is further described in subsection 4.6.3 to generate pattern anomalies. The generated pattern anomalies are randomly injected into the data. In the experiments, we apply both the baseline approach presented in subsection 4.4.2 and our quality-aware approach presented in section 4.5

on this data with the injected anomalies. We then compare the precision, recall, F1-score, and accuracy of each approach in detecting these injected anomalies.

Finally, the elbow method was used to determine the number of clusters for the k-means clustering method. It is a heuristic method used to determine the number of clusters k for clustering algorithms [Dangeti, 2017].

4.6.3 . Anomaly Generation and Injection

For our experiments, we have implemented a customized automated anomaly generator capable of generating anomalies of the various types defined earlier for MCS environments in section 4.2. Our anomaly generator is automated as shown in Figure 4.3. It takes as input the data file as a csv file and the following parameters: the window size W, which is also the size of the pattern anomaly to be injected. The second parameter, *anomalies-per*, is the percentage of anomalies to be injected into the data. Given our anomaly generator creates the values of the anomalies randomly, this means that for each attempt of anomaly generation, the values are different than the previous run even if we had the same anomalies percentage in parameter *anomalies-per*. The last parameter comprises the specification of the percentage of each type of anomaly to be injected. In our experiments, we only injected pattern anomalies. Hence the percentages for noise and point anomalies were set to zero.

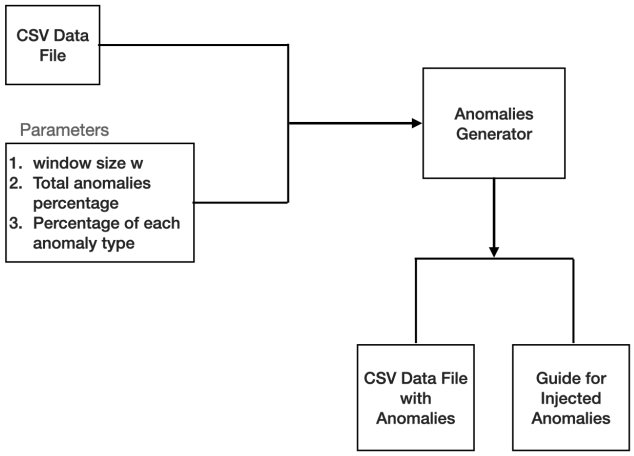

Figure 4.3: The Anomaly Generator.

The anomaly generator provides anomaly values randomly. The indices at which the anomalies will be injected are also chosen randomly with the setting that generates a value within a specific range and allows for a random generated

value to be repeated more than once. The pattern anomaly length is defined by the *window size* parameter and has randomly generated values within the possible range of values of the studied pollutant. The range of accepted values of the studied element is also an input to the generator.

The last feature of our anomaly generator is that it outputs 2 files. The first output file contains the original data records with the injected anomalies as a csv file. The second output file is a guide that indicates where the anomalies were injected, their types in case different types of anomalies were injected, and their corresponding values. This guiding file is mandatory for the experiments in order to be able to track which anomalies were successfully detected and which ones were not.

4.6.4 . Evaluation Metrics

To assess the usefulness of our quality-aware approach to anomaly detection on time series in mobile crowdsensing environments, we evaluate the results using the following metrics:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$Precision$ measures the proportion of positive predictions that are actually positive. Precision is very useful in contexts with a high cost of false positives.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

$Recall$ measures the proportion of positive cases that the classifier correctly predicted. The recall is useful when a high cost is associated with false negatives.

The $F_1 - score$ is a trade-off between precision and recall. It is computed as their harmonic mean. F1-Score is needed when a balance between Precision and Recall is desired. In our experiments, we use $F_1 - score$ to evaluate our results because there is no additional cost associated with either false positives or false negatives. Both have to be equally avoided. Originally, F1-score is Fβ-score where β controls giving higher importance to either precision or recall depending on the context of the application and which metric has higher importance than the other.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$FPRate = \frac{FalsePositive}{FalsePositive + TrueNegative}$$

$FP - rate$ is the false positivity rate that tells us the ratio of false positives over the total number of ground truth negatives.

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative}$$

Accuracy is the ratio of the correctly identified points (positives and negatives) in the dataset. Accuracy is important to characterize the number of the model's predictions that were correct. This metric is relevant in most applications.

4.6.5 . Results

In this subsection, we present the results of our experiments. The first experiment is done on dataset 1 comprising 32 000 data records. We experiment on both the baseline approach and our quality-aware approach injecting different percentages of anomalies in the dataset. We first try injecting 1%, 2%, 3%, and 4% of anomalies in dataset 1, and compare the F1-Score and accuracy achieved by both the baseline approach and our approach. Figure 4.4 shows the F1-score achieved by both approaches. Our approach outperforms the baseline approach. However, the achieved results from both approaches both fall below 0.2, especially for the baseline approach. The baseline approach has a very poor F1-Score for all of the tested anomaly percentages injected. The main reason is that the precision was low, with a value of 0.1. This means that the number of false positives is significantly greater than the number of true positives.

Figure 4.5 shows the achieved accuracy of this experiment by both the baseline

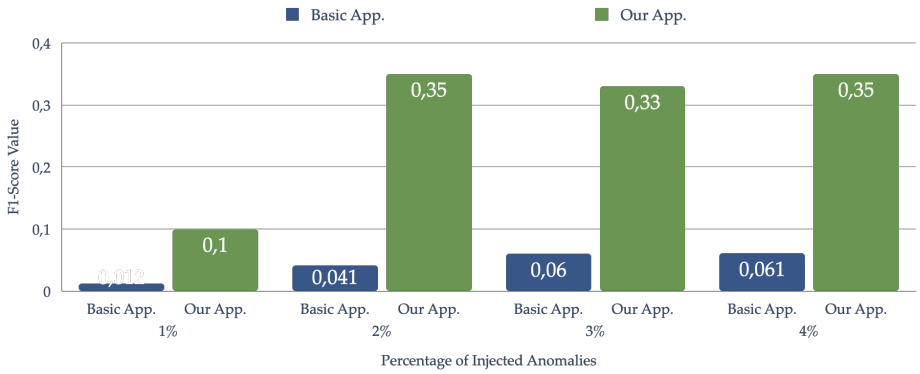

Figure 4.4: F1-Score achieved by both baseline approach and our approach after injecting 1%, 2%, 3% and 4% of anomalies on dataset 1.

and our approach. We can see that the values of accuracy for this experiment are higher than the F1-Score for both approaches. When 2% of anomalies are injected, our approach achieves a high accuracy value of 0.94 while the baseline approach is 0,72 accurate. We also observe that the accuracy achieved by both approaches for 3% of injected is similar, with a slightly higher value achieved by our approach. The accuracy of the baseline approach later decreases with 4% injected anomalies, while our approach scores a higher accuracy of 0.9.

We then examine, on the same dataset, the impact of changing the values of other parameters of the algorithm, such as the *window size w*, the threshold δ used

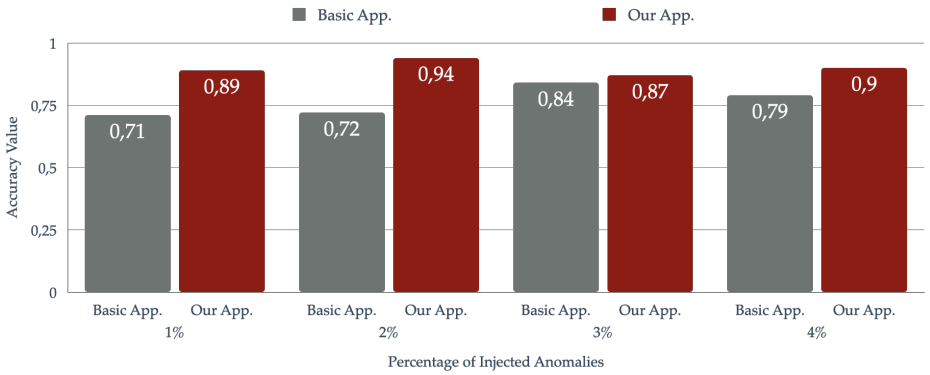

Figure 4.5: Accuracy values achieved by both baseline approach and our approach after injecting 1%, 2%, 3% and 4% of anomalies on dataset 1.

to determine whether an anomaly score of a subsequence means it is anomalous or not, and the number of clusters k. Figure 4.6 shows the values of precision,

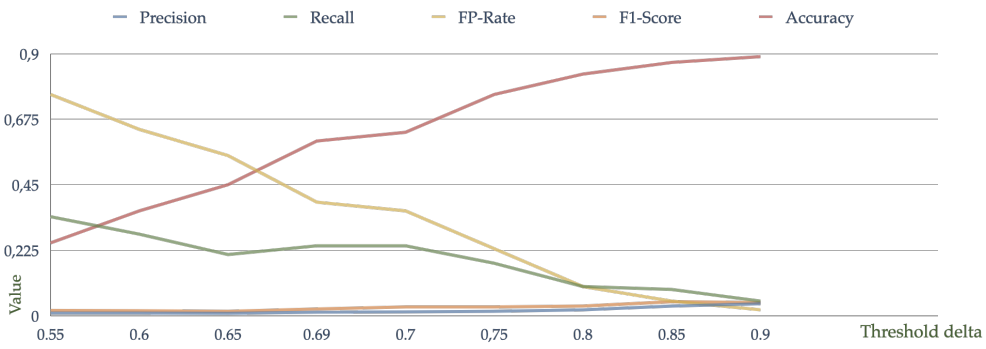

Figure 4.6: Quality of the baseline approach on dataset 1 with $w = 7$, k=4, and 4% injected anomalies.

recall, F1-Score, FP-rate, and accuracy achieved by the baseline approach when 4% of anomalies were injected in dataset 1, with window size w=7, and the number of clusters k=4. The graph shows the values of the evaluated metrics as a function of the δ threshold. The precision values result in a lower F1-score than those obtained by our approach in Figure 4.7. We notice that the FP-rate is at its highest value, in both figures, at the lowest anomaly threshold δ, and then it decreases as the δ threshold increases. This is an intuitive behavior because, with a higher threshold, a smaller number of sequences are detected as anomalous.
The FP-rate and F1-Score are inversely correlated, meaning for example, as FP-rate decreases, the F1-score increases. Nonetheless, the F1-score still falls below 0.1 with this experiment. However, both approaches obtain high accuracy scores with a noticeable difference between the baseline and our approach where

our approach shows higher accuracy scores than those of the baseline one.

Figure 4.8 and Figure 4.9 show the results of the studied metrics for the same data setting as the latter experiment, but we increase the window size to be w=8 and keep the number of clusters k=4. Figure 4.8 shows similar scores to that of Figure 4.6. Both precision and F1-Score show low scores that fall below 0.1. Recall is higher with values between 0.5 and 0.25 for δ threshold between 0.55 and 0.68, but lower than 0.25 for δ value $>$ 0.68. However, the results of our approach in Figure 4.9 show that the precision and recall had their highest values at $\delta = 0.7$ with values of 0.31 and 0.42 respectively. The accuracy at this δ threshold is high with a value of 0.84.

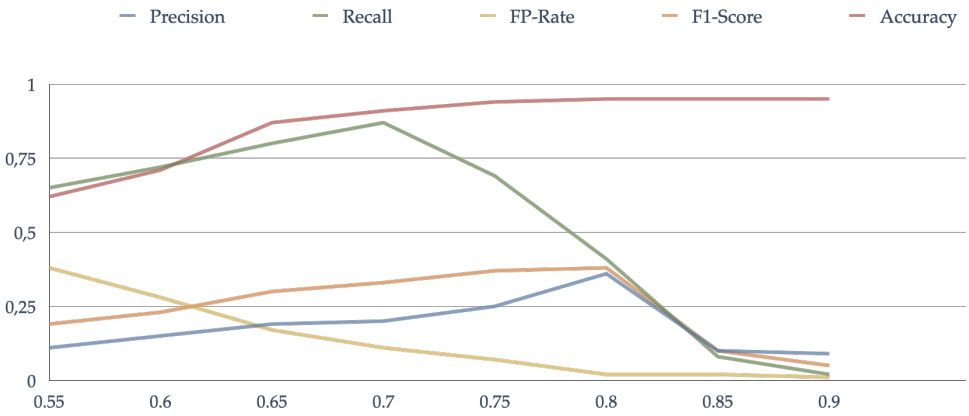

Figure 4.7: Quality of our approach on dataset 1 with $w = 7$, k=4, and 4% injected anomalies.

Figure 4.10 and Figure 4.11 show the results of the studied metrics achieved by the baseline and our approach for the same data settings as the latter setup, but with window size W= 9 and number of clusters k = 5. Figure 4.10 shows the results achieved by the baseline approach, which shows similar values to that of Figure 4.8 where precision and F1-Score are below 0.1 for all δ thresholds. The accuracy is higher than 0.5 for $\delta \geq 0.73$. Figure 4.11 here shows the scores of the evaluated metrics by our approach for window size W= 9 and the number of clusters k = 5. We observe that recall, precision, and F1-Score are at the highest achieved values 0.92, 0.31, and 0.48 respectively at threshold $\delta = 0.8$. Accuracy is also at the highest levels with a value of 0.93 for the same δ threshold.

The achieved accuracy results are above 0.5 for this experiment starting at a threshold value of $\delta = 0.7$. However, the F1-Score results of the baseline approach were always below 0.1. Our approach works better for this data setup.

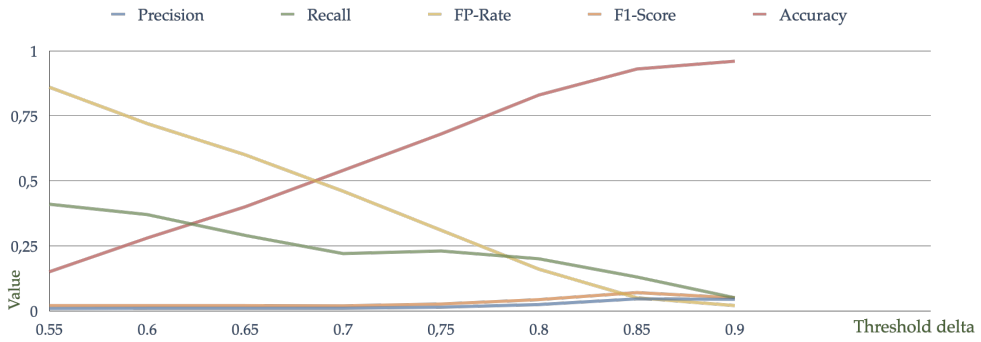

Figure 4.8: Quality of the baseline approach on dataset 1 with $w = 8$, k=4, and 4% injected anomalies.

This leads us to the second type of experiments conducted on dataset 2. On this dataset, we examine the F1-Score and accuracy achieved after injecting different percentages of anomalies into the dataset 5%, 10%, 15%, 20%, and 25%.

Figure 4.12 shows the achieved F1-scores of both the baseline and our approach on dataset 2 for 5%, 10%, 15%, 20%, and 25% injected anomalies. We notice that for small percentages, the baseline approach works better for dataset 2 than dataset 1. We also observe that with higher injected anomaly percentages, both the baseline approach and our approach score significantly higher F1-score. This graph also shows that our approach always scores a higher F1-Score than the baseline approach. We notice a higher F1-score achieved by the baseline approach as we increase the percentage of anomalies in the dataset. The highest F1-score is achieved by the baseline approach at 25% of injected anomalies. On the other hand, our approach shows higher F1-scores when the injected percentage of anomalies was 10%, 15%, 20%, and 25%, having the highest value of 0,82 at 25% injected anomalies compared to 0.68 F1-score by the baseline approach. As for the accuracy, Figure 4.13 shows improved accuracy scores for both the baseline approach and our approach. The accuracy of the baseline approach is at its highest with a value of 0.55 for 25% injected anomalies, while the highest achieved by our approach, with a value of 0,92, was at 10% injected anomalies.

Finally, knowing that the anomalies were generated and injected randomly in the dataset, we validate our achieved results by supporting them with two additional executions of the anomalies injection percentages, with the same experimental setup, to confirm the achieved results.

Figure 4.14 and Figure 4.15 show the F1-Score of the two additional executions of anomalies injection to confirm the achieved results. We notice a slight change

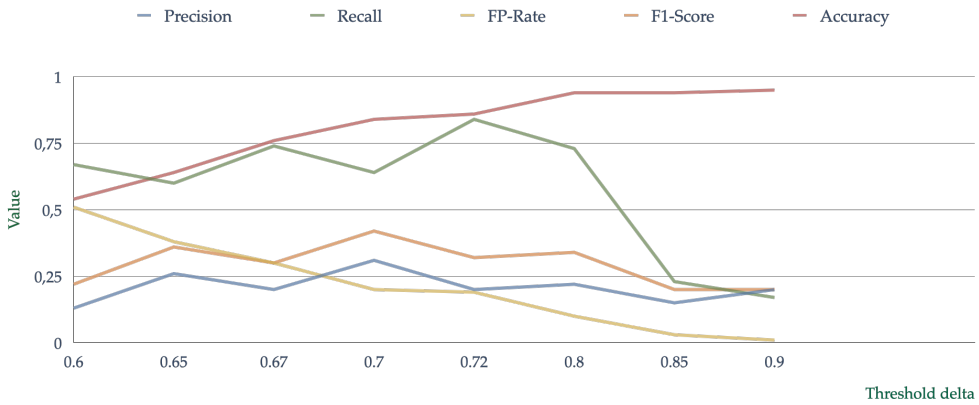

Figure 4.9: Quality of our approach on dataset 1 with $w = 8$, k=4, and 4% injected anomalies.

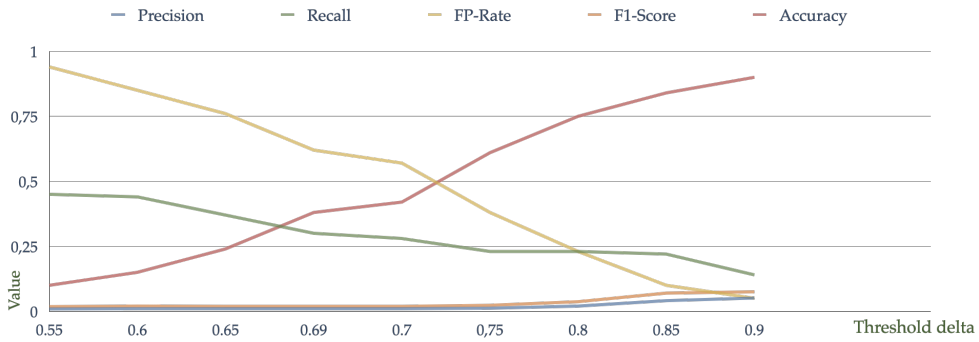

Figure 4.10: Quality of the baseline approach on dataset 1 with $w = 9$, k=5, and 4% injected anomalies.

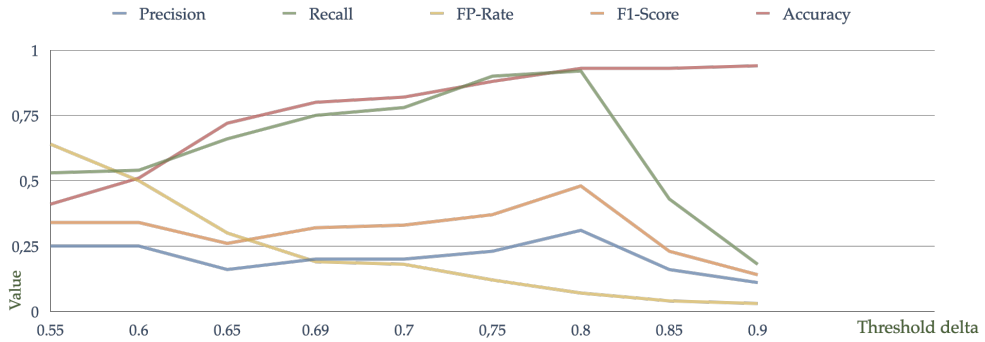

Figure 4.11: Quality of our approach on dataset 1 with $w = 9$, k=5, and 4% injected anomalies.

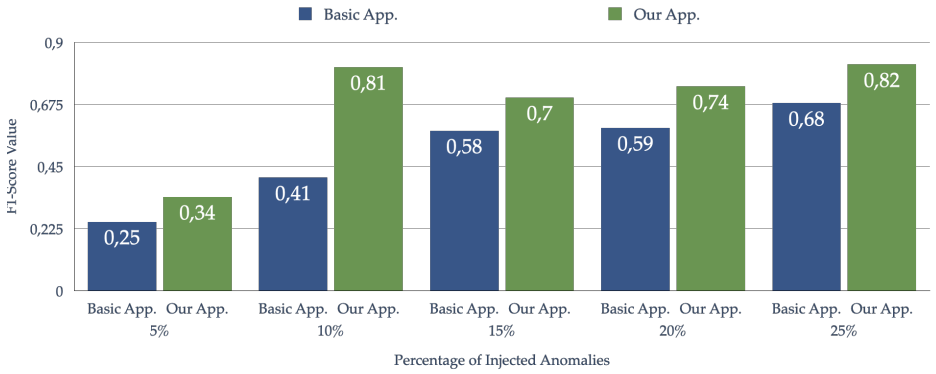

Figure 4.12: F1-Score for baseline and our approach with different percentages of anomalies on dataset 2.

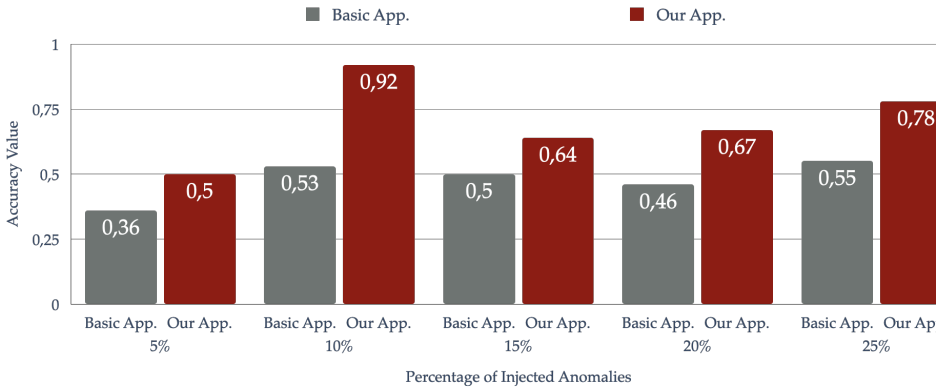

Figure 4.13: Accuracy for baseline and our approach with different percentages of anomalies on dataset 2.

in the achieved F1-Score of some experiments but with a very small margin. Our approach maintains higher F1-scores than the baseline approach for all of the percentages of injected anomalies.

Likewise, Figure 4.16 and Figure 4.17 show the accuracy achieved by the baseline approach and our approach over the second and the third executions of the various percentages of anomalies injected. The graphs clearly show that our approach maintains higher accuracy scores when compared to the baseline approach. We also notice an insignificant change in the achieved accuracy among the different executions. Furthermore, the results are consistent among the three different executions, which confirms the validity of the achieved results on accuracy and F1-score.

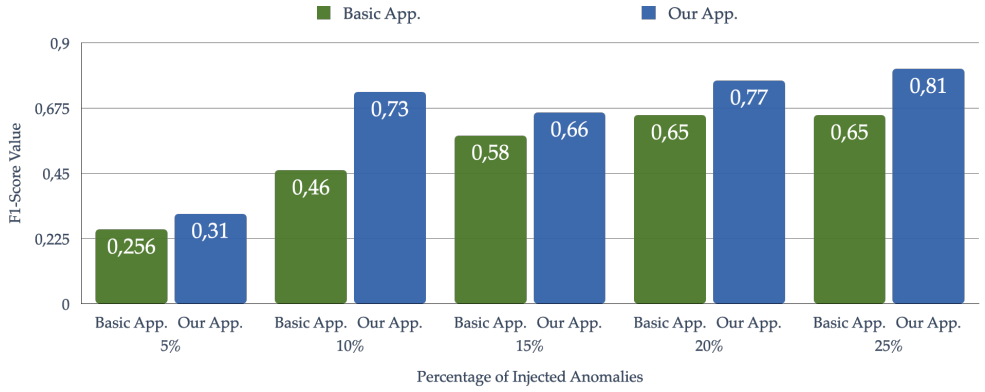

Figure 4.14: F1-Score of the 2^{nd} execution of anomalies injection for baseline and our approach with different percentages of anomalies on dataset 2.

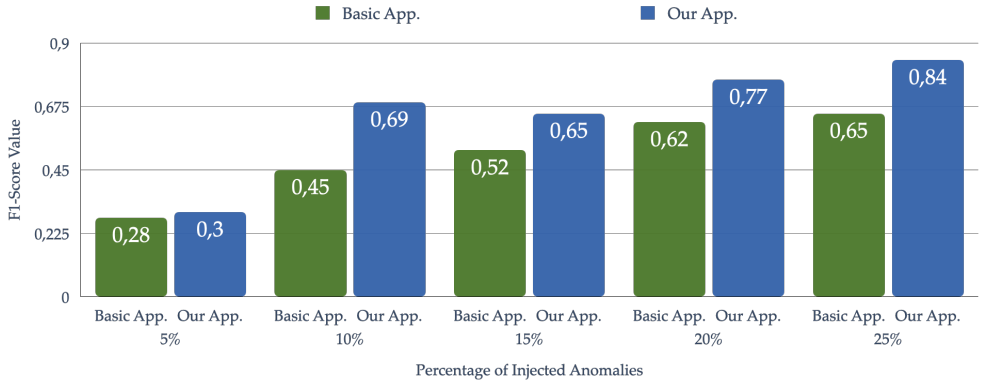

Figure 4.15: F1-Score of the 3^{rd} execution of anomalies injection for baseline and our approach with different percentages of anomalies on dataset 2.

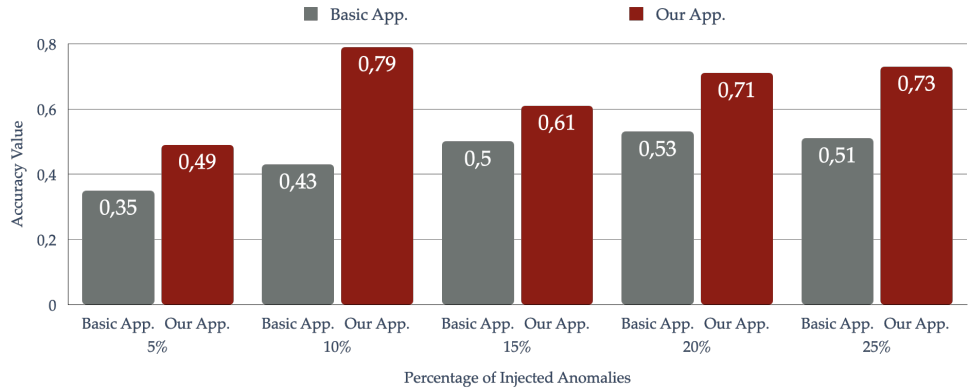

Figure 4.16: Accuracy of the 2^{nd} execution of anomalies injection for baseline and our approach with different percentages of anomalies on dataset 2.

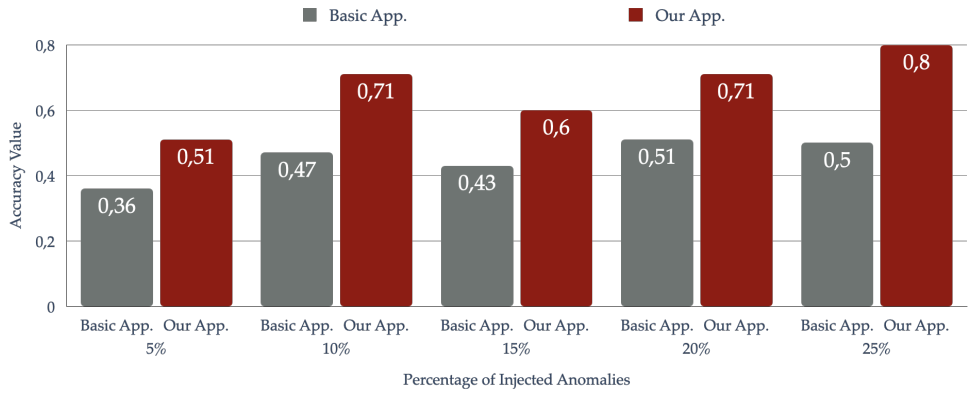

Figure 4.17: Accuracy of the 3^{rd} execution of anomalies injection for baseline and our approach with different percentages of anomalies on dataset 2.

4.6.6 . Discussion

Our experiments have shown significant advantages of our quality-aware approach over the baseline approach in identifying anomalies for time series data. For injected percentages of anomalies less than or equal to 4%, our approach performed remarkably better than the baseline approach with a highest F1-score of 0.5 with our approach compared to a value less than 0.1 for the baseline approach. The accuracy for this experiment type where the window length, the number of clusters, and the δ threshold were varied, vary inversely for both approaches with the FP-rate. Precision scored the lowest values among the studied metrics both approaches for this setup with a highest value of 0.3 with our approach compared to a value less than 0.1 for the baseline one.

Experiments using dataset 2 with the increasing injected percentages of anomalies, 5% through 25%, worked remarkably better on both the baseline and our approach. However, despite significantly improving precision and F1-Score, the accuracy achieved by the baseline approach on dataset 1 was better than when executed on dataset 2. The accuracy achieved by the baseline approach on dataset 2 was around 0.5 compared to values always greater than 0.5 by our approach. F1-scores achieved on both approaches were considerably improved on dataset 2 with a highest value of 0.84 by our approach at 25% injected anomalies compared to a highest value of 0.68 by the baseline approach at the same percentage of injected anomalies.

4.7 . Conclusions

In this chapter, we presented a quality-aware anomaly detection approach that uses the quality of the sensor and the context on time series data collected within the context of mobile crowdsensing environments (MCS). We defined anomalies for the MCS context and defined the types of anomalies in this context: $noise$, $point$, and $pattern\ anomalies$. Our proposed approach improves the detection of anomalies by grouping similar subsequences together using the quality of the sensor and some contextual information. The similarity function takes into account the measurement values of the subsequence as well as both the quality and the context of each subsequence. In our approach, we define the notion of *subsequence quality* that helps group subsequences of similar qualities together and compute the quality of the clusters. We also define the notion of *cluster quality* that was used to assign an anomaly score to the subsequences in the dataset.

We showed through the experiments that an anomaly detection approach driven by quality and contextual information always performs better than an approach that works without taking the quality and the context into account.

We achieved an F1-score that is almost twice better using our approach than the baseline approach at the experiment setup with 10% added anomalies. An accuracy of 0.92 was achieved with our approach compared to 0.53 with the baseline one at this percentage of anomalies injected in the data. Our approach heavily relies on the quality of the measuring sensors. The quality of the sensor is an aggregate of several quality dimensions that assess various facets of the sensors and hence is a value of the quality of the measurements taken by the sensor at the time of the evaluation.

For future works, it would to investigate approaches that assess the uncertainty of the estimations of the anomaly detection model. Uncertainty estimation in the context of anomaly detection involves quantifying and propagating the uncertainty associated with the detection results. Anomaly detection algorithms typically assign anomaly scores or probabilities to data points, indicating their likelihood of being anomalous. Uncertainty estimation goes further by providing a measure of confidence or uncertainty associated with these scores. This can provide users with a better understanding of the reliability of detected anomalies and support decision-making processes. For datasets with a large number of features or dimensions, we must explore anomaly detection techniques that work for high-dimensional data. In such datasets, the number of features can be comparable to or even exceed the number of data points, posing unique challenges for anomaly detection algorithms. Addressing such a challenge requires a combination of appropriate preprocessing techniques, specialized algorithms, feature selection methods, and possibly some domain knowledge.

We can also investigate techniques that can leverage knowledge learned from one domain or dataset to improve anomaly detection in another related domain or dataset. This includes developing transfer learning or domain adaptation methods to mitigate the challenge of limited labeled data in new domains.

5 - Towards Quality-Aware Sensor Data Analytics in MCS

5.1 . Introduction

In mobile crowdsensing environments, mobile, low-cost sensors are used to collect a huge number of data measurements for some measurable elements, such as the pollution level in the air, traffic congestion, etc. This data has to be reconciled in order to be cleaned and stored to be made available for the users in a centralized repository comprising all the collected measurements ready for analysis.

When analyzing this integrated data, the users may be manipulating measurements of varying levels of quality due to potential sensor malfunctions, manufacturing defects, and sometimes due to the usage of the sensor by the carrier. As a result, the data provided by these sensors can have a low quality which could lead to indicators of poor quality.

In the previous chapters, we have dealt with two quality dimensions. The first is data completeness, where we characterize and propose metrics to measure completeness factors. Furthermore, we enhance data completeness by implementing an approach that generates missing values while considering the quality of the measuring sensors. The second is accuracy, where we propose a quality-aware anomaly detection to detect pattern anomalies in the data. Likewise, other existing metrics could be relevant for mobile crowdsensing environments. In this chapter, our objective is to take quality into account when using the data provided by the sensors. Our proposal in this chapter is a first step towards a quality-aware data analytics pipeline in which data quality plays a central role in computing indicators and analysis.

Given a set of metrics suitable for assessing the quality of the data in mobile crowdsensing environments, our goal is twofold: the first is to compute indicators of higher quality by considering the quality of the input data, and the second is to use the quality of the data to characterize the quality of an indicator computed based on this data. For example, in an air quality monitoring context, if we want to determine the pollution levels of particulate matter in a certain city, we would like to take into account the variation of quality among the sensors and give more importance to the ones that are of higher quality. Assuming this set of metrics, we are also interested in providing centralized storage of this information about quality in order to exploit it along with the data during the analysis process.

The authors of [Berrahou et al., 2015, Boulil et al., 2013] have integrated quality in their data models, and incorporated this data about the quality in their analysis. Likewise, we follow a similar approach in representing the data quality information in our model, and integrate the data quality in the computation of indicators based on it.

In chapter 3, we have introduced a multidimensional data model representing data provided by sensors within a mobile crowdsensing environment. In this chapter, we enrich this model with information about quality, allowing the representation of quality-related aspects in mobile crowdsensing environments.

The quality of the indicators depends on the quality of the underlying data. Hence, we propose to enrich basic aggregation operators with the quality of this data. The first operator computes a quality score for each data measurement of a given measured element taken by a sensor. To deliver higher-quality insights using queries and aggregation operators on the extended model, we then propose two quality-aware aggregation operators that consider quality while computing the aggregate. The first aggregation operator uses quality as a weight to aggregate data measurements. The second operator filters out some data measurements depending on their levels of quality.

We propose a method that computes the quality of an aggregate given the quality of the input data used to compute this aggregate. Our approach utilizes the quality scores of the data measurements used in the aggregation, in order to assess the quality of the aggregate. The goal of this method of qualifying the computed indicators is to provide the level of quality of an aggregated so the users can take this quality into account during their decision-making process.

The remainder of this chapter is organized as follows. Section 5.2 presents the quality multidimensional representation of data in the MCS environment. Section 5.3 discusses the two proposed quality-aware aggregation operators. In section 5.4, we present an approach to characterize the quality of computed aggregates. Finally, we conclude the chapter in section 5.5.

5.2 . A quality multidimensional model of MCS data

In this section, we present the representation of data quality for sensor data in a mobile crowdsensing environment. We recall the multidimensional data model presented in chapter 3 that introduced the representation of sensor data in this context. This model consisted of a fact table *Measurement* that describes a data measurement in the MCS environment. This measurement describes a *physical element* in real-life measured by a *sensor* unit. The other dimensions of this

measurement are the *user* that is carrying the sensor, the *time* at which the measurement is taken, and the geographic *location* where the measurement was captured.

This section presents our representation of data quality information within a mobile crowdsensing environment in Figure 5.2. This multidimensional data model enriched with quality values provides the data consumer with the quality values corresponding to different aspects of a sensor. The sensor quality value fact table has the following dimensions: *time*, *sensor*, and *sensor quality*.

The content of the fact table depends on the quality goals of the application. Similarly to the work defined in [Berti-Équille et al., 2011], a data quality analyst defines the quality goals that are refined and decomposed into a set of quality questions. The answer to a quality question is defined by choosing and refining a quality factor that best characterizes the question and a set of quality metrics that are appropriate to measure this factor [Berti-Équille et al., 2011]. The values resulting from these evaluations are called quality values in our work, similarly to the meta-model defined in [Berti-Équille et al., 2011], these quality values represent the result of executing a measurement method, for a measurable object, at a given point in time.

Example 5.2.1. Suppose a data quality analyst manages the quality of sensor data measuring air pollution in a specific area. Assume the analyst has ten sensors measuring black carbon and wants to detect those that underperform because their level of completeness is deteriorating. A possible quality scenario would be to assess sensor completeness every three days. Hence, after one month, there would be ten values of data completeness quality factor for each sensor. In Figure 5.1, we show the instances of the fact table of the data values collected for one sensor unit s_1 for the data completeness quality factor.

Sensor Quality Value ID	Quality Factor ID	Sensor ID	Time ID	Sensor Quality Value
1	Completeness	s_1	2023-05-01 12:00:00+00	0.8
2	Completeness	s_1	2023-05-04 12:00:00+00	0.75
3	Completeness	s_1	2023-05-07 12:00:00+00	0.36
4	Completeness	s_1	2023-05-10 12:00:00+00	0.51
5	Completeness	s_1	2023-05-13 12:00:00+00	0.78
6	Completeness	s_1	2023-05-16 12:00:00+00	0.91
7	Completeness	s_1	2023-05-19 12:00:00+00	0.6
8	Completeness	s_1	2023-05-22 12:00:00+00	0.65
9	Completeness	s_1	2023-05-25 12:00:00+00	0.82
10	Completeness	s_1	2023-05-28 12:00:00+00	0.87

Figure 5.1: Instances of the fact table showing the recorded quality values of the completeness factor of one sensor unit.

Each sensor can be assessed using one or several quality factors at different timestamps. Evaluating one quality factor QF_k, for instance, accuracy would result in a quality value instance $qv_{k_{i,j}}$ of ID qid, quantifying the accuracy of a sensor unit s_i at a specific timestamp t_j. This instance is denoted by (qid, QF_k, s_i, t_j, $qv_{k_{i,j}}$).

Example 5.2.2. O Consider a mobile crowdsensing application deployed in an urban area to monitor black carbon levels at various locations throughout the city. Assume we have two sensors s_1 and s_2 that captured measurements of black carbon at timestamps t_1, t_2, and t_3. Sensor s_1 captured the measurements {(1,t_1), (5,t_2), (18,t_3)} and sensor s_2 captured the measurements {(1,t_1), (10,t_2), (15,t_3)}.
In order to meet the quality assessment goals for the data, this data is continuously evaluated. The assessed quality factors and their corresponding values are stored in the quality multidimensional data model. Hence, values of the accuracy, completeness, and consistency of the data coming from sensors s_1 and s_2 at different timestamps are stored in the model as instances shown below.
(1, 'Consistency', s_1, t_1, 0.22), (2, 'Completeness', s_1, t_2, 0.67), (3, 'Accuracy', s_1, t_3, 0.91)
(4, 'Consistency', s_2, t_1, 0.85), (5, 'Completeness', s_2, t_1, 0.40), (6, 'Accuracy', s_2, t_1, 0.73).

The model presented in Figure 5.2 focuses on representing the quality factors and values of the sensors capturing a physical element in the fact table *SensorQualityValue* and its dimension *SensorQualityFactor*. The *SensorQualityValue* fact table and its relevant dimensions are described as follows.

Sensor Quality Value Fact Table

The fact table describes the values of the measured quality factors of each sensor unit at a certain timestamp. Each instance of this fact table represents a quality value corresponding to the evaluation of a given factor for a given sensor at a given timestamp. For example, a quality value of 0.7 could be the quality value for the consistency quality factor of a sensor s_1 at timestamp t_1.

Sensor dimension

The sensor dimension represents the sensor unit that captures measurements of the physical element. In this table, the sensor has a sensorType attribute and a sensorName. It could also contain other attributes describing the characteristics of the sensor, such as its weight, or its frequency. The sensor is associated with the fact table SensorQualityValue where each instance in this fact table corresponds to a quality factor for a specific sensor unit.

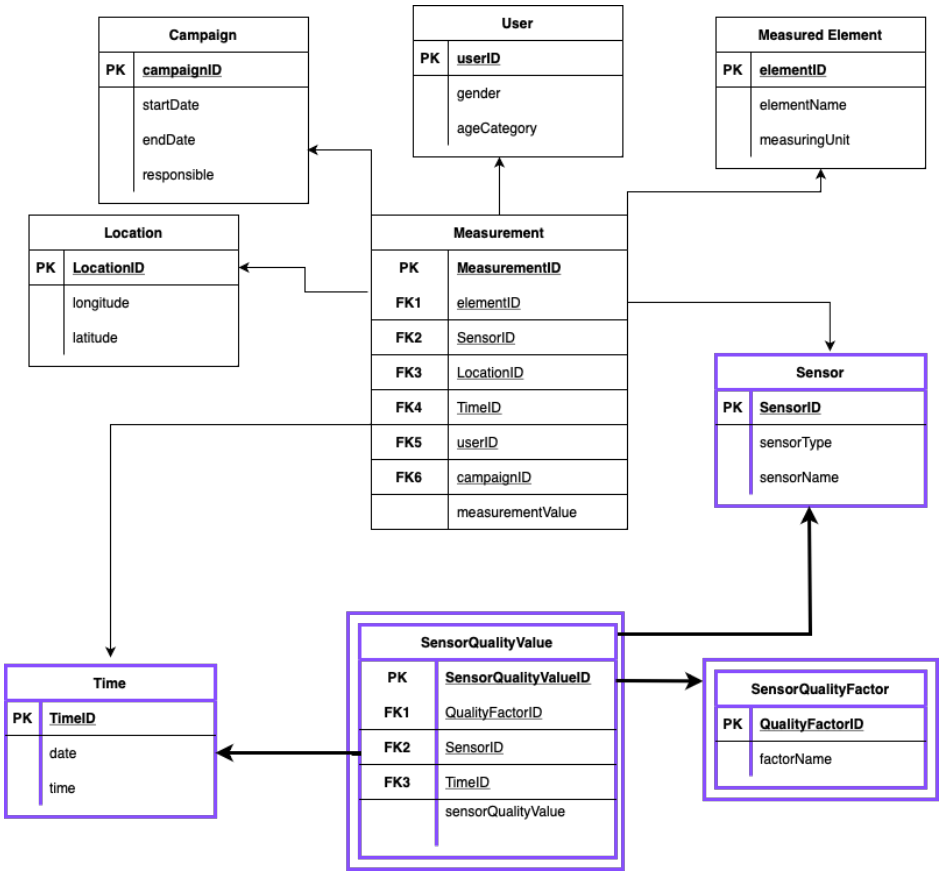

Figure 5.2: Quality Multidimensional Model.

Sensor quality factor dimension

This dimension represents the quality factors that describe a quality aspect of the sensor. Instances of this dimension are quality factors that characterize data coming from a sensor such as completeness, comparison with reference data, consistency, etc. For example, a factor could be data completeness, which indicates the sensor completeness described in subsection 3.4.1.

Time dimension

Another dimension of interest is the time. It is composed of two attributes: time and date. It will be used to characterize the time of an assessment of a quality

factor of some sensor.

Quality goals are set on sensors to assess their performances by evaluating some quality factors that are relevant to them. For instance, if we are interested in evaluating the accuracy of the data coming from a sensor over the month of May, a data quality goal is set to evaluate the accuracy of this sensor everyday during this month. A sensor could have several quality values of the same quality factor evaluated at different timestamps.

5.3 . Quality-aware Aggregation Operators

In multidimensional data models, aggregation operators can be executed while taking into consideration data from some dimensions. An aggregation operator is conventionally defined as a function that is applied to a collection of tuples and returns a single value [Damiani and Spaccapietra, 2006],[López et al., 2005]. Examples of existing aggregation functions are min, max, avg, sum, $spatial\ fusion\ in\ geographical\ systems$, and others. The authors of [Bimonte et al., 2006] defined aggregation operators that support geographical data in the Spatial OLAP context.

In this section, we present our approach on quality-aware aggregation operators. The approach integrates data quality in the aggregation operator. This proposal of quality-aware aggregation operators is based on the multidimensional model defined in section 5.2 that integrates the data quality aspects.

Mobile sensors are vulnerable to errors and are likely to face points of failure, which leads to some collected measurements by these sensors that may not be reliable. Manufacturing defects in a sensor can cause significant disruptions on the functioning of the sensor and can therefore lead to collected data measurements by this sensor that are of poor quality. Reliability concerns are brought to question on the data coming from some sensors due to dissipated power of the sensor unit [Ergun et al., 2021], and several other causes. Hence, the data aggregation operators need to be aware of the quality of the data being used. We propose aggregation operators that could allocate a higher weight to sensors that are more reliable during a certain period of time. The idea is to give a higher importance to measurements coming from high-quality sensors and less importance to those coming from low-quality ones. We illustrate this in the following.

Example 5.3.1. Assume we have 2 data measurements of a given measurable physical element, such as the level of pollution, from sensors s_1 and s_2 having values of 10 and 20 at timestamp t_j respectively. Suppose sensor s_1 has an assessed quality score of 0.95, and sensor s_2 has a quality score of 0.25 at timestamp t_k that is the closest in time to timestamp t_j than other registered quality scores. If we want to compute the average of the measurements at timestamp t_j, the

conventional average aggregation operator would compute the average of the value of both values, which is 15. However, if we take quality into account, the value of the average would be closer to that of the high-quality sensor s_1 rather than a value right in the middle between s_1 and s_2. Hence, the new value of this aggregation would be closer to 10 than 15.

In the following subsections, we present our method to compute a global quality score of the sensor. We also present our quality-aware aggregation operators extended while taking data quality into account.

5.3.1 . Computing a sensor quality score

This subsection discusses the quality score of a single data measurement in mobile crowdsensing environments. In Figure 5.2, we have presented the representation of quality in the multidimensional model. In this section, we discuss how we could use the model to compute a global quality score of a given sensor at a specific timestamp.

We consider that the quality of a given measurement is characterized by the quality score of the sensor that has taken this measurement. This means that the data measurements have the same quality as the sensors taking them around the acquisition time. We recall that in our work, we consider that the quality of the sensor is characterized by some quality factors related to different aspects of the sensor that are device-related, usage-related, or related to comparison with reference data.

A data measurement taken by some sensor is associated with the registered evaluations of the quality factors that are the closest in time to it. This means that the closest value $qv_{k_{i,j}}$ of the quality factor QF_k in time, is associated with the data measurement $v_{i,j}$.

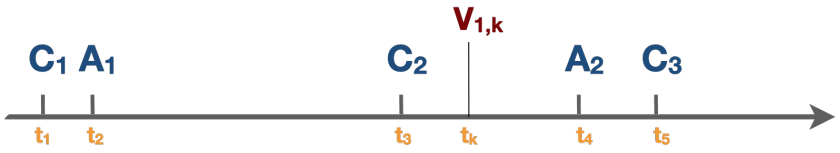

Figure 5.3: An example of quality assessments of completeness and accuracy of a sensor s_1 at different timestamps.

Example 5.3.2. Assume we are interested in the completeness and accuracy quality factors, and that these two factors have been evaluated according to the timeline presented in Figure 5.3 Suppose the completeness quality factor of

sensor s_1 has been evaluated at timestamps t_1, t_3, and t_5 with quality values $C_1 = 0.2, C_3 = 0.85$, and $C_1 = 0.9$ respectively. The accuracy factor of the same sensor is also evaluated at two other timestamps, t_2, and t_4, with the quality values $A_1 = 0.85$ and $A_2 = 0.55$ respectively.
The timestamp t_3 of the completeness value C_2, and the timestamp t_4 of the accuracy value A_2 are the closest to the timestamp t_k of measurement $v_{i,k}$. Hence, the completeness value C_2=0.2 and the accuracy value A_2=0.55 are associated to measurement $v_{i,k}$.

This data quality information can be retrieved from the multidimensional data model using the three tables Sensor, SensorQualityValue, and SensorQualityFactor. For example, if our multidimensional data model is implemented as a relational model, we could retrieve the quality values that are the closest to a measurement taken at a certain timestamp using the following SQL query shown in Figure 5.4.

Query Editor | Query History

```
1  SELECT qv.sensorQualityValue, time, s.sensorType
2  FROM "SensorQualityValue" as qv
3  INNER JOIN "SensorQualityFactor" as qf on(qv.QualityFactorID = qf.QualityFactorID)
4  INNER JOIN "Sensor" as s on (s.SensorID = qv.SensorID)
5  order by abs(extract(epoch from (time - '2023-05-23 10:10:00')))
6  limit 1;
```

Figure 5.4: Query to retrieve the closest quality value to the timestamp of a measurement assuming a relational implementation of our multidimensional model.

Depending on the considered quality goal set, some factors could be relevant to an aggregation task, and others would not. We assume that the quality values correspond to the quality factors in a set $F = \{QF_1, \ldots, QF_n\}$ where n is the number of chosen relevant quality factors. For some quality goals, the quality value of a single quality factor could be relevant, and for other quality goals, a set of selected quality factors could be relevant.

The sensor global quality score is a value that summarizes the quality of a set of predefined quality factors F. The quality factors in the set F are determined according to the quality goals of a specific application. If the quality values of the factors are all normalized, it is possible to compute an aggregate that represents the global quality score of a single data measurement.

Definition 5.3.1. (Sensor Global Quality Score) Consider a set of quality values that correspond to the quality factors in a predefined set $F = \{QF_1, \ldots, QF_n\}$. The quality score $Qscore_{i,j}$ of sensor s_i at timestamp t_j is computed as follows.

$$Qscore_{i,j}(F) = \frac{\sum_{k=1}^{n} qv_{k_{i,j}}}{n} \tag{5.1}$$

Where $qv_{k_{i,j}}$ is the value of a quality factor QF_k of sensor s_i assessed at a timestamp that is the closest to t_j considering only the quality values related to a quality factor in F, n is the number of relevant quality factors.

Example 5.3.3. Consider the sensors s_1 and s_2 and their quality values presented in example 5.2.2. The global quality score of both sensors s_1 and s_2 at timestamp t_3 can be computed as follows:
$Q_{score_{1,3}}$ = avg(0.22 + 0.67 + 0.91) = 0.60
and $Q_{score_{2,3}}$ = avg(0.85 + 0.4 + 0.73) = 0.66.
Hence, the global quality score of a measurement taken by sensor s_1 at timestamp t_3 is 0.6, and the global quality score of a measurement taken by sensor s_2 at timestamp t_3 is 0.66.

5.3.2 . Extending existing aggregation operators

Some approaches have proposed extensions of aggregation operators. For example, the authors of [Bimonte et al., 2006] defined operators for the Spatial OLAP context that support spatial data aggregations. The authors of [Berrahou et al., 2015] extend existing aggregation operators with data quality values. During the aggregation, they use the quality values to filter out data measurements that have a quality value below a certain threshold. Similarly, in this section, we present our quality-aware approach that extends existing aggregation operators using the data quality. A quality-aware aggregation operator is one where the data quality values are considered during the computation of an aggregate. The idea behind this proposal is to improve the quality of the aggregation results by taking quality into account. This means that measurements with higher quality will be assigned higher importance and, thus, weights than those with lower quality.

We propose two different aggregation methods that take data quality into account. The first method is for the average aggregation operator. The second method can be applied to any aggregation operator, such as the average, sum, etc.

Weighting-based Aggregation

In this aggregation approach, each measurement is weighted by its quality value. This would lead to high-quality measurements being assigned a higher weight than those with lower quality. We consider that the quality of a data measurement is in fact the global quality score of the sensor taking that measurement evaluated at the closest timestamp to the time of the measurement. We recall that in our work, we consider that sensor quality is defined by quality values computed using device-related, usage-related, and reference-related metrics. The weight is the global quality score of a measurement that is either computed as the value of a single factor or as an aggregate of several quality values corresponding to several quality factors.

Definition 5.3.2 (Weighting-based Aggregation)**.** The weighting-based aggregation operator ϕ_1 on the set of measurements $X_N = \{v_1, \ldots, v_N\}$ collected by multiple sensors s_k with $k = \{1, \ldots, n\}$ and having the corresponding global quality scores $\{Qscore_1, \ldots, Qscore_N\}$ is computed as follows.

$$\phi_1(X_N) = \frac{\sum_{i=1}^{N} v_i \times Qscore_i}{\sum_{i=1}^{N} Qscore_i}$$

Where $\phi_1(X_N)$ is the quality aggregation operator, N is the total number of measurements, v_i is a data measurement, and $Qscore_i$ is the global quality score of data measurement v_i.

We illustrate an aggregation of this method with the following example that shows how quality is integrated into the aggregation.

Example 5.3.4. Consider the measurements collected by sensors s_1 and s_2 and their corresponding quality values presented in example 5.2.2. The average levels of black carbon in the city at a certain timestamp t_3 can be computed using quality aggregation operator. To this end, we aggregate the available measurements from sensors s_1 and s_2 at timestamp t_3 weighted by their quality:

$$\phi_1 = \frac{(18 \times 0.6) + (15 \times 0.66)}{0.6 + 0.66} = 16.4$$

Filtering-based Aggregation

The work of [Berrahou et al., 2015] uses the data quality to filter out the data measurements whose quality value falls below a predefined threshold. Likewise, we propose a filtering-based aggregation operator that filters out the data measurements that have a global quality score that is below a certain predefined threshold δ. We recall that we consider that the quality score of a data measurement is the quality score of the sensor taking that measurement.

A quality aggregation operator ϕ_2 is an aggregation that uses quality to filter out the data measurements that have a global quality score below a certain predefined threshold δ. We define a quality-aware aggregation operator ϕ_2 on the set of measurements X_N as follows.

Suppose a set of measurements $X_N = \{v_1, \ldots, v_N\}$ collected by multiple sensors s_k with $k = \{1, \ldots, n\}$ having the corresponding global quality scores $\{Qscore_1, \ldots, Qscore_N\}$.

Definition 5.3.3 (Filtering-based Aggregation)**.** The filtering-based aggregation operator ϕ_2 applied on the set of measurements $X_N = \{v_1, \ldots, v_N\}$ considering a quality threshold δ and an aggregation function Λ is computed as follows.

$$\phi_2(X_N) = \Lambda(v_i \times f_i) \qquad \text{where } \begin{cases} f_i = 1 & \text{if } Qscore_i \geq \delta \\ f_i = 0 & \text{otherwise} \end{cases} \tag{5.2}$$

Where $Qscore_i$ is the global quality score of measurement v_i, δ is a predefined threshold, and Λ is an aggregation function, such as max, sum, average, etc.

5.4 . Assessing the quality of an aggregate

We have discussed in the previous sections quality-aware aggregation operators that take data quality into account. In this section, we discuss our approach to assess the quality of an aggregate.

Data measurements in a dataset could come from different sensors with different levels of quality. This implies that some data measurements that the aggregation is based on, could have very low levels of quality and others very high ones. This raises the question on the reliability of the aggregate. Providing the quality of the aggregate along with the aggregated value can help the end users better characterize the level of trust that can be drawn on these indicators. For example, assume an aggregate with a very low-quality score, important decisions would not be made based on this aggregate. Likewise, given an aggregate of a very high quality, we are more confident in making important decisions based on this aggregate.

Example 5.4.1. Suppose 2 sets of 5 data measurements each, $d_1 = \{m_1, m_2, m_3, m_4, m_5\}$ and $d_2 = \{m_6, m_7, m_8, m_9, m_{10}\}$ both measuring the same element and having the following quality scores respectively {0.1, 0.05, 0.15, 0.11, 0.08} and {0.9, 0.85, 0.95, 1, 0.88}. This means that in d_1, measurement m_1 has a quality score of 0.1. Assume we use the same aggregation operator to aggregate the values of each set of measurements and obtain 2 aggregates. The quality of the aggregate resulting from set d_1 will be much lower than that of set d_2 because the measurements in set d_1 have quality scores that are much lower than those in set d_2.

The idea is to propose an approach that computes the quality of an aggregate based on the quality of the measurements that have been used to compute it. We propose an aggregation method that reuses the quality score of each measurement that was proposed in subsection 5.3.1. The quality of an aggregate depends on the quality of the underlying measurements.

Suppose a set of measurements $X_N = \{v_1, \ldots, v_N\}$ collected by multiple sensors s_k with $k = \{1, \ldots, n\}$ having the corresponding global quality scores $\{Qscore_1, \ldots, Qscore_N\}$.

Definition 5.4.1. The quality of an aggregate computed by an aggregation method $\phi(X_N)$ that computes an aggregate of the set of measurements $X_N = \{v_1, \ldots, v_N\}$ having the corresponding global quality scores

$\{Qscore_1, \ldots, Qscore_N\}$, is computed as follows.

$$Q_{\phi(X_N)} = \frac{\sum_{i=1}^{N} Qscore_i}{N} \tag{5.3}$$

Where $Qscore_i$ is the global quality score of a measurement v_i.

Example 5.4.2. Consider the measurements collected by sensors s_1 and s_2 and their corresponding quality values presented in example 5.2.2. Assume that the aggregation operator ϕ computes the aggregate value of the average levels of black carbon a human is exposed to during timestamps t_1 and t_2 with a value of 20. The underlying data is composed of a set X_N of 4 measurements coming from sensors s_1 and s_2 of respective quality values of 0.6 and 0.66 at the time.
According to our proposed evaluation technique, the quality of this aggregate $Q_{\phi(X_N)}$ is:
sum(0.6 + 0.6 + 0.66 + 0.66)/4 = 0.63

5.5 . Conclusion

In this chapter, we illustrated how the data quality can be used to either compute an aggregate taking quality into account or to assess the quality level of a given aggregate. We enriched the multidimensional data model by integrating information about the quality factors and their corresponding quality values at a given timestamp and for a given sensor. These quality values are recorded at specific timestamps. The quality of the measurement is determined by the value that is closest in time to the timestamp of the measurement. We proposed an aggregation operator that computes a global quality score of a data measurement.

We have also proposed two quality-aware aggregation operators that integrates the quality of the data during the aggregation. The first operator is weighting-based that uses the data quality as a weight to compute an aggregate. The second operator is filtering-based that filters out data measurements according to their quality scores. We also introduce a method to characterize the quality of an aggregate. This method computes the quality of computed aggregates by relying on the quality of the measurements that this aggregate is performed on to help data consumers make quality-informed decisions.

The chapter proposes preliminary ideas in order to implement a quality-based data analytics pipeline. In future works, it would be interesting take data quality into account in more complex data manipulations, such as in data mining algorithms and machine learning pipelines.

Moreover, exploring how different stakeholders perceive and prioritize data quality in decision-making processes would help tailor data analytics pipelines to meet specific quality-related user needs and preferences.

6 - Conclusions and Perspectives

Data quality issues for sensor data in mobile crowdsensing environments have recently gained attention and have been addressed by several reseach works. With the rise of smart cities and connected objects in almost every aspect of our daily lives, data scientists spend a significant part of their time ensuring a data quality level suitable for analytics. Numerous recent works have studied quality problems related to sensor data, and have proposed solutions to assess and improve data quality. Ensuring good data quality has become a crucial process to accomplish good-level indicators for decision-making in mobile crowdsensing environments.

In this chapter, we first summarize the contributions of this thesis. We then present some perspectives and possible directions for future works.

6.1 . Summary of our contributions

In this thesis, we have targeted several major data quality issues in mobile crowdsensing environments. We first addressed data completeness issues in this context. We defined a set of quality factors for data completeness suitable for mobile crowdsensing environments. We then proposed metrics for the evaluation of the various data completeness factors identified. We have also addressed the improvement of data completeness, and we proposed a quality-aware data imputation approach based on existing imputation techniques that takes data quality into account during the generation of the missing values. In the second part of our work, we have also addressed issues related to the presence of anomalies in sensor data. To this end, we have designed a quality-aware anomaly detection approach to better detect pattern anomalies while considering the quality of data in mobile crowdsensing environments. We finally proposed a first contribution towards quality-aware sensor data analytics. We have proposed quality-aware aggregation operators and a method to assess the quality of an aggregate.

We have studied the limitations of existing quality factors for mobile crowdsensing environments. We proposed three data completeness factors suitable for this context: sensor completeness, spatial completeness, and temporal completeness. These three factors allow for capturing facets of data that are specific to mobile crowdsensing environments. We have tested the metrics to evaluate each of the proposed completeness factors on real data. We introduced the definition of sensor quality and presented a way to compute a quality score for each sensor. We also extended three existing data imputation techniques and presented a way to take into account the quality of the sensors in the generation process in order to improve the quality of the generated values. We have also evaluated our approach on real data. The experiments showed that integrating quality helps improve the

generation of missing values.

We presented a quality-aware anomaly detection approach that uses quality to compute the anomaly scores of sequences in the time series. We defined anomalies for mobile crowdsensing environments and identified three types of anomalies. We presented a novel anomaly score computation based on the quality of the existing data. We have also incorporated some features about the context in our anomaly detection approach. The experiments showed that our quality-based approach achieves a significant improvement in the F1-score and accuracy compared to the baseline approach. Our approach achieves an F1-score of 0.81 and 0.92 accuracy compared to 0.41 F1-score and 0.53 accuracy achieved by the baseline approach.

Finally, we have proposed a first step towards quality-aware sensor data analytics. We have defined an approach to compute a global quality score of a data measurement in mobile crowdsensing environments. The computation of the quality score can be parameterized by the set of quality factors that have to be considered, which depends on the specific quality goal of the application. We have proposed two types of aggregation operators that take data quality into account while computing the aggregates, a weighting-based operator and a filtering-based one. Given a computed aggregate, we also propose an assessment method that evaluates the quality of this aggregate based on the quality of the input data used in the aggregation.

6.2 . Future works

In this section, we outline the perspectives and propose future directions to further advance the research in data quality for mobile crowdsensing environments and explore new opportunities for potential contributions.

In the same way as we have characterized data completeness for mobile crowdsensing environments, we could explore other dimensions of data quality which capture a quality facet that is not covered by existing dimensions. An interesting dimension in this context is the quality of the information related to the spatial coordinates of data measurements. Many works have studied positioning technologies [Evennou, 2007], [Kammoun, 2016]. It would be useful to characterize the quality of the spatial positioning of data measurements and to and propose metrics to assess this quality.

The evaluations of the data completeness factors showed that the performance could vary significantly from one sensor unit to the other. The performance of the sensors also degrade over time. Another perspective we could explore in future works is to use our approach to evaluate the quality of the data generated by

a sensor regularly over intervals of time, then use these evaluations to learn the pattern of how the quality of sensors change over the time in order to predict the quality of new coming sensors. We are also interested in studying the impact of the usage of the sensor by its carrier on the quality of the data generated by this sensor by studying the quality of sensors from same manufacturer, type, measuring the same element, and in the same city over a period of time.

To improve the completeness of sensor data, we proposed a quality-aware data imputation approach that was limited to imputing a single data measurement at a time in time series. This means that our approach assumed that only one data measurement would be missing at one timestamp and not a huge chunk of consecutive missing values. However, sensors could sometimes blackout and hence, lose massive chunks of their data. In future works, we would like to study the imputation of big chunks of data by exploring approaches that exploit nearby sensors while considering the time and the context of this missing data. For example, if a sensor that was placed inside a metro station loses data of one whole week. We could explore techniques that leverage data from other nearby sensors that were inside the metro station at the time of the blackout, and data from other sensors that captured the same element at metro station to generate the missing data.

In order to detect anomalies for massive amounts of data due to the rise of smart cities and the proliferation of mobile sensors in MCS environments, one opportunity that we can investigate is efficient and scalable anomaly detection. In order to cope with the scalability problems posed by massive datasets such as the computational complexities on the anomaly detection models or the processing time, we could leverage distributed computing. One way is to extend our anomaly detection approach by parallelizing the processing of the different steps on distributed machines in order to handle bigger datasets in less time.

Finally, we have proposed in chapter 5 two different quality-aware aggregation operators. It would be interesting to investigate automated generic aggregation operators that could be customized and instantiated according to different preferences of the users. For example, if the user stores information about their quality goals and preferences, the generic aggregation operators would automatically take the quality preferences of this user profile, set the relevant quality factors to the quality goals, and integrate their values in the aggregation accordingly.

A - Further Evaluations of Data Completeness

In this chapter, we present the remainder of the results performed for improving data completeness in chapter 3. The datasets and the methodology of the experiments were presented in chapter 3. We present the results of our experiments on both extended techniques SVDImpute and ST-MVL proposed to generate missing values. For ST-MVL, we only present the results of the sub-techniques of this approach.

Results of SVDImpute extension on both datasets for filtering using quality-above-threshold

Table A.1: Results of Quality above threshold 0.55 for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	30%	64%	6%	23.5
[100, 0, 0]	23%	18%	59%	9.6
[0, 100, 0]	60%	27%	13%	1.6
[0, 0, 100]	34%	51%	15%	18.7
[60, 20, 20]	36%	24%	38%	2.3
[20, 60, 20]	67%	31%	2%	8.3
[20, 20, 60]	37%	60%	3%	24.1
[40, 40, 20]	41%	39%	20%	16.8
[40, 20, 40]	28%	64%	8%	23.1
[20, 40, 40]	49%	50%	1%	13.3

Table A.1 shows the results of the filtering method of SVDImpute considering only sensors above the threshold of 0.55 on dataset 1. The weight configuration [20,60,20], mainly highlighting the sensor completeness facet showed the best improvement results with 67% also validated by the weight configuration [0,100,0] that also shows 60% improvement, a relatively high improvement percentage. The [20,60,20] and [0,100,0] weight configurations showed 31% and 27% worsening percentages with RMSE of 8.3 and 1.6 respectively, indicating that the error difference between the generated value with our extension and the actual value is not huge despite being less accurate than that generated with the baseline approach. The two weight configurations also showed 2% and 13% of unchanged measurements respectively.

Both [33,33,34] and [40,20,40] weight configurations showed the worst results of 64% of worsened values with, respectively, 23.5 and 23.1 RMSE error metric

values.

The weight configuration [20,20,60] shows a 60% of worsened values, which is also a high percentage. This weight configuration gives the major importance to the third facet which is the correlations from the comparison with reference data. This may indicate that for this quality threshold (0.55), the third facet in combination with the other facets induce poor results.

For the weight configuration [0,0,100], keeping only the third facet, 51% of the measurements were worsened. This means that for this weight configuration, keeping only sensors with a quality threshold above 0.55, almost 50% of the values were worse with our extensions.

The wight configuration [100,0,0] shows the most unchanged measurements percentage with a value of 59% meaning the IPI index alone does not have a huge impact on the improvement of the performance of the technique.

The rest of the weight configurations highlighting the IPI index [60,20,20], or the IPI index and the sensor completeness equally [40,40,20], showed slightly more improved values than worsened ones.

Table A.2: Results of Quality above threshold 0.65 for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	-	-	-	-
[100, 0, 0]	25%	13%	62%	7343
[0, 100, 0]	-	-	-	-
[0, 0, 100]	40%	57%	3%	19.5
[60, 20, 20]	55%	43%	2%	12.3
[20, 60, 20]	62%	36%	2%	6.3
[20, 20, 60]	44%	53%	3%	20
[40, 40, 20]	53%	47%	0%	5.5
[40, 20, 40]	41%	57%	2%	21.2
[20, 40, 40]	50%	48%	2%	6.6

Table A.2 shows the results of the filtering method of SVDImpute considering only sensors above the threshold of 0.65 on dataset 1. The weight configurations [33,33,34] and [0,100,0] have no data because, for these weight configurations, extended SVDImpute does not have measurements from sensors with quality above the 0.65 threshold.

The weight configuration [20,60,20] showed the best result with 62% improved measurements. It also had 36% of measurements worsened with RMSE 6.3 and 2% of unchanged measurements. This means that giving sensor completeness the bigger weight increases the percentage of improved values.

The two weight configurations [0,0,100] and [40,20,40] show the worst results with 57% worsened values. We notice that for both configurations, the sensor completeness facet was either zero or assigned the smallest weight. These configurations also show 41% and 40% of improved values respectively, and 3% and 2% of unchanged values.

The weight configuration [100,0,0] has the biggest unchanged measurements: 62%. Yet, it produced more improved measurements than worsened with 25% improved and 13% worsened. This weight configuration gives major importance to IPI index, which is the quality facet that evaluates the sensors at the beginning of the campaign, so the big unchanged measurements percentage may mean here that this facet alone does not have a great impact on the technique.

Table A.3: Results of Quality above threshold 0.75 for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	-	-	-	-
$[100, 0, 0]$	37%	62%	1%	6.6
$[0, 100, 0]$	-	-	-	-
$[0, 0, 100]$	-	-	-	-
$[60, 20, 20]$	-	-	-	-
$[20, 60, 20]$	-	-	-	-
$[20, 20, 60]$	-	-	-	-
$[40, 40, 20]$	-	-	-	-
$[40, 20, 40]$	-	-	-	-
$[20, 40, 40]$	-	-	-	-

Table A.3 shows the results of the filtering method of SVDImpute considering only sensors above the threshold of 0.75 on dataset 1. For this experiment, we notice that only the weight configuration [100,0,0] has data, meaning that all the others did not have sensor measurements with quality above the 0.75 threshold. This weight configuration is [100,0,0], which resulted in 62% of the values worsened, 6.6 RMSE value, 37% improved, and 1% of the values remained unchanged.

Table A.4 shows the results of the filtering method of SVDImpute considering only sensors above the threshold of 0.45 on dataset 2. The percentage of improved

Table A.4: Results of Quality above threshold 0.45 for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	26%	14%	60%	697.9
[100, 0, 0]	25%	13%	62%	750.9
[0, 100, 0]	42%	17%	41%	627179
[0, 0, 100]	51%	38%	11%	12.8
[60, 20, 20]	26%	13%	61%	101.4
[20, 60, 20]	19%	12%	69%	181.8
[20, 20, 60]	45%	37%	18%	435976
[40, 40, 20]	26%	14%	60%	247986
[40, 20, 40]	44%	30%	26%	174.3
[20, 40, 40]	43%	32%	25%	10.7

measurements is greater than that of worsened for all the 10 experimented weight configurations. As there are a lot more data in this dataset, there are more spike values from the sensors resulting in an amplified RMSE value.

The best improvement result is produced by the weight configuration [0,0,100], where the major importance is given to the facet of sensor quality that compares the measurements from the sensors to reference data. 38% of the measurements were worsened with this configuration with an RMSE value of 12.8. This configuration resulted in the highest worsened number of measurements.

Table A.5: Results of Quality above threshold 0.55 for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	45%	38%	17%	8.1
[100, 0, 0]	25%	13%	62%	1.7
[0, 100, 0]	42%	17%	41%	162612
[0, 0, 100]	48%	42%	10%	12.5
[60, 20, 20]	26%	14%	60%	117.7
[20, 60, 20]	48%	31%	22%	254050
[20, 20, 60]	50%	41%	9%	13.2
[40, 40, 20]	44%	31%	25%	10.6
[40, 20, 40]	44%	39%	17%	1650
[20, 40, 40]	49%	36%	15%	86.3

Table A.5 shows the results of the filtering method of SVDImpute considering only sensors above the threshold of 0.55 on dataset 2. The percentages of improve-

ment were higher than those worsened for all the 10 studied weight configurations. However, there is no major difference between the percentages of the values that improved and those that worsened.

The best improvement result achieved with the weight configuration [20,20,60] is 50%, where the correlations from the comparison with reference data are given the highest importance. At the same time, it has almost the highest rate of worsened measurements, yet the number of improved values is still higher than those worsened. The RMSE value is 13.2, which is also relatively not high.

The highest percentage of worsened value is achieved in the weight configuration [0,0,100]. The RMSE is 12.5, which is relatively not high.

The two weight configurations [100,0,0] and [60,20,20] which either take only the IPI index facet or give the IPI index the greatest importance, show the highest percentage of unchanged measurements 62% and 60% respectively, indicating that this facet, once more, does not have a huge impact on the technique.

Table A.6: Results of Quality above threshold 0.75 for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	-	-	-	-
$[100, 0, 0]$	55%	39%	6%	7.3
$[0, 100, 0]$	-	-	-	-
$[0, 0, 100]$	-	-	-	-
$[60, 20, 20]$	-	-	-	-
$[20, 60, 20]$	-	-	-	-
$[20, 20, 60]$	-	-	-	-
$[40, 40, 20]$	-	-	-	-
$[40, 20, 40]$	-	-	-	-
$[20, 40, 40]$	-	-	-	-

Table A.6 shows the results of the filtering method of SVDImpute considering only sensors above the threshold of 0.75 on dataset 2. Only the $[100, 0, 0]$ weights set configuration has results for this quality threshold. It shows 55% improvement and 39% with a 7.3 RMSE error value. Even though more measurements were improved with our approach than worsened, the worsened percentage is relatively high.

Table A.7: Results of Top 50% sensors for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	61%	30%	9%	7.3
[100, 0, 0]	53%	37%	10%	7.3
[0, 100, 0]	62%	28%	10%	3.7
[0, 0, 100]	57%	33%	10%	11.2
[60, 20, 20]	63%	27%	10%	8.3
[20, 60, 20]	64%	25%	11%	4.2
[20, 20, 60]	60%	30%	10%	8.2
[40, 40, 20]	62%	27%	11%	4.2
[40, 20, 40]	61%	30%	9%	7.9
[20, 40, 40]	61%	27%	12%	8.2

Results of SVDImpute extension on both datasets for filtering using 50% top-quality sensors

Table A.7 shows the results of the filtering method of SVDImpute considering only top 50% sensors on dataset 1. In this experiment, we also observe high improvement percentages as with the top 40% sensors for this setting. Most of the weight configurations show 60% and above improved values. The worsened percentages range between 25% and 30%.

The two weight configurations $[100, 0, 0]$ and $[0, 0, 100]$ are the only configurations with improved values less than 60%.
The weight configuration with best improved values is once again $[20, 60, 20]$ giving the most importance to the sensor completeness facet.

Table A.8 shows the results of the filtering method of SVDImpute considering only top 70% sensors on dataset 1. Generally, our extension works worse with all the weight configurations in this experiment than the previous two percentages 40% and 50$ because most of the weight configurations lead to less than 60% improved values. However, the weight configuration $[20, 60, 20]$ has the highest improved values: 65%. The second best set of weights are $[0, 100, 0]$ with only sensor completeness facet, and $[20, 40, 40]$; with weights distributed evenly between sensor completeness and correlation coefficients and a less weight given to IPI index.

The worsened values achieved by the different weight configurations ranged between 18% with $[20, 40, 40]$ weight configuration and 37% for the weight configuration $[0, 0, 100]$.
The unchanged values were in the same range for all the weight configurations ranging between 11% and 19%.

Table A.8: Results of Top 70% sensors for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	54%	33%	12%	9.7
$[100, 0, 0]$	46%	28%	16%	10.2
$[0, 100, 0]$	63%	24%	14%	28.3
$[0, 0, 100]$	47%	37%	16%	13
$[60, 20, 20]$	56%	32%	12%	10
$[20, 60, 20]$	65%	22%	13%	5
$[20, 20, 60]$	52%	36%	12%	12.9
$[40, 40, 20]$	62%	23%	15%	7.2
$[40, 20, 40]$	53%	36%	11%	12.7
$[20, 40, 40]$	63%	18%	19%	9.5

Table A.9: Results of Top 90% sensors for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	51%	30%	19%	11.3
$[100, 0, 0]$	45%	32%	23%	165.7
$[0, 100, 0]$	55%	28%	17%	23.3
$[0, 0, 100]$	46%	35%	19%	14.4
$[60, 20, 20]$	51%	29%	20%	11.2
$[20, 60, 20]$	55%	26%	19%	6.3
$[20, 20, 60]$	48%	33%	19%	14.1
$[40, 40, 20]$	58%	22%	20%	6.1
$[40, 20, 40]$	50%	29%	21%	13.6
$[20, 40, 40]$	50%	32%	18%	11.8

Table A.9 shows the results of the filtering method of SVDImpute considering only top 90% sensors on dataset 1. All of the achieved improved values by the different weight configurations range between 45% and 58%; 58% being the highest with the weight configuration $[40, 40, 20]$.

The percentage of unchanged measurements was the highest at 23% for the $[100, 0, 0]$ weight configuration.
The range of the values worsened almost remains the same among the different weight configurations. This indicates that for 90% of the top performing sensors, which is almost all the sensors there is, more values remain unchanged while less values are improved. This conclusion makes sense as 90% of top performing sensors

is not too far away from 100% which is the baseline approach.

Table A.10: Results of Top 40% sensors for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	53%	27%	20%	24.1
[100, 0, 0]	53%	29%	18%	23.6
[0, 100, 0]	50%	28%	22%	23.7
[0, 0, 100]	52%	28%	20%	24.4
[60, 20, 20]	53%	27%	20%	24.8
[20, 60, 20]	51%	29%	20%	23.5
[20, 20, 60]	52%	28%	20%	24.3
[40, 40, 20]	52%	27%	21%	23.5
[40, 20, 40]	54%	26%	20%	25.2
[20, 40, 40]	55%	25%	20%	191610

Table A.10 shows the results of the filtering method of SVDImpute considering only top 40% sensors on dataset 2. The range of improved values is small for this experiment as it ranges between 51% and 55% only. This means that the different weight configurations studied had a similar impact on the technique with small differences.

Generally, more improvement was done to the measurements imputation than worsened. However, the worsened percentage is relatively comparable to that improved ones where it ranged between 15% and 28%.

Table A.11: Results of Top 50% sensors for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	52%	26%	22%	3.8
[100, 0, 0]	51%	28%	21%	23.2
[0, 100, 0]	51%	26%	23%	16792498
[0, 0, 100]	50%	28%	23%	25.9
[60, 20, 20]	51%	26%	23%	25.8
[20, 60, 20]	50%	27%	23%	26
[20, 20, 60]	51%	24%	25%	25.7
[40, 40, 20]	52%	22%	26%	25.9
[40, 20, 40]	52%	20%	28%	25.6
[20, 40, 40]	60%	15%	25%	19.1

Table A.11 shows the results of the filtering method of SVDImpute considering only top 50% sensors on dataset 2. The achieved percentage of improved values were almost the same between the top 40% and top 50% for this data setting, except for the weight configuration $[20, 40, 40]$ which shows 60% of improved values of the measurements. Only 15% of the measurements are worsened for this weight configuration and 25% of the measurements remained unchanged.

The percentages of unchanged measurements slightly rose from selecting top 40% to top 50% and the percentages worsened slightly diminished.

Table A.12: Results of Top 90% sensors for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	42%	17%	41%	100.2
$[100, 0, 0]$	41%	19%	40%	35.9
$[0, 100, 0]$	42%	16%	42%	96
$[0, 0, 100]$	40%	19%	41%	120.7
$[60, 20, 20]$	42%	17%	41%	196456767
$[20, 60, 20]$	43%	17%	40%	112.2
$[20, 20, 60]$	40%	18%	42%	97.3
$[40, 40, 20]$	43%	17%	40%	140.4
$[40, 20, 40]$	41%	18%	41%	111.2
$[20, 40, 40]$	44%	18%	38%	114

Table A.12 shows the results of the filtering method of SVDImpute considering only top 90% sensors on dataset 2. As we take top 90% performing sensors, which is close to 100% which is the baseline approach, we notice that the improved and unchanged values are almost the same in most of the weight configurations.
The percentages of the worsened values are relatively low, ranging between 17% and 19% compared to improved values ranging between 40% and 44%.

Results of ST-MVL sub-techniques

Table A.13 shows the results of the evaluation of the proposed extension of IDW, a sub-technique of ST-MVL, on dataset 1. The four different weight configurations $[0, 100, 0]$, $[20, 60, 20]$, $[40, 40, 20]$ and $[20, 40, 40]$ show 70% of improved values. These weight configurations mainly focus the majority of the weight on the sensor completeness facet.

The weight configuration $[100, 0, 0]$ showed the worst results where it produced only 16% of the measurements improved while 69% of them were worsened. It also shows 15% of them remained unchanged. The weight configuration $[0, 0, 100]$ also

Table A.13: Results of IDW for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	68%	18%	14%	4.7
[100, 0, 0]	16%	69%	15%	5.1
[0, 100, 0]	70%	17%	13%	3.5
[0, 0, 100]	43%	44%	13%	5.2
[60, 20, 20]	66%	19%	15%	4.8
[20, 60, 20]	70%	16%	14%	4.2
[20, 20, 60]	59%	27%	14%	4.8
[40, 40, 20]	70%	17%	13%	4.6
[40, 20, 40]	60%	25%	15%	4.9
[20, 40, 40]	70%	17%	13%	4.5

shows poor results with 44% of the measurements worsened and 43% improved.

When taking only one facet at a time, the sensor completeness is the only facet that shows good results. The other two facets show poor results when taken alone. The percentages of measurements unchanged are generally below 16%.

Table A.14: Results of IDW for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	36%	25%	39%	42.2
[100, 0, 0]	28%	30%	42%	40.1
[0, 100, 0]	30%	30%	40%	35.9
[0, 0, 100]	31%	19%	50%	50.9
[60, 20, 20]	34%	25%	41%	41.2
[20, 60, 20]	34%	7%	59%	40.2
[20, 20, 60]	33%	19%	58%	45.5
[40, 40, 20]	37%	6%	57%	40.6
[40, 20, 40]	33%	6%	61%	43.3
[20, 40, 40]	35%	6%	59%	43

Table A.14 shows the results of the evaluation of the proposed extension of IDW, a sub-technique of ST-MVL, on dataset 2. The improvement values of IDW for dataset 2 is low with a highest improvement value of 37%. The percentages range between 28% and 37%. However, only 6% of the measurements were worsened and 57% remained unchanged, which means a significant number of the measurements were not affected with the extension.

The worsened measurements range between 6% and 30%. Hence, even though our extensions did not show high improvement percentages, they showed a little worsening of the measurements with most of the weight configurations.

Table A.15: Results of UCF for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
$[33, 33, 34]$	10%	90%	0%	3.1
$[100, 0, 0]$	15%	84%	1%	2.4
$[0, 100, 0]$	8%	92%	0%	3.5
$[0, 0, 100]$	7%	93%	0%	3.5
$[60, 20, 20]$	11%	89%	0%	2.3
$[20, 60, 20]$	8%	92%	0%	3.3
$[20, 20, 60]$	8%	92%	0%	3.3
$[40, 40, 20]$	10%	90%	0%	3
$[40, 20, 40]$	10%	90%	0%	3
$[20, 40, 40]$	8%	92%	0%	3.2

Table A.15 shows the results of the evaluation of the proposed extension of UCF, a sub-technique of ST-MVL, on dataset 1. UCF mostly performed poorly for this data setting with more than 90% of measurements worsened on average for all the weight configurations studied.

The percentages of improvement are insignificant with a highest recorded value of 15% by the $[100, 0, 0]$ weight configuration. The unchanged measurement are almost zero for this experiment.

Even though around 90% of the measurements were worsened for this setting, the RMSE error is very small. This shows that the imputation value by our extension was slightly further than the actual one than baseline approach.

Table A.16 shows the results of the evaluation of the proposed extension of UCF, a sub-technique of ST-MVL, on dataset 2. This technique performed better for dataset 2 than for dataset 1. Nevertheless, the improvement percentages were still less than those worsened. The improvement percentages ranged between 39% and 44% for weight configurations $[0, 0, 100]$ and $[0, 100, 0]$ respectively.

The highest improvement achieved considered one facet only, which is the sensor completeness with the weight configuration $[0, 100, 0]$. However, this configuration had 54% of the measurements worsened which is high in comparison with the improved measurements. The highest worsening percentage achieved was

Table A.16: Results of UCF for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	42%	56%	2%	19.9
[100, 0, 0]	43%	55%	2%	19.8
[0, 100, 0]	44%	54%	2%	20
[0, 0, 100]	39%	56%	5%	20.1
[60, 20, 20]	43%	53%	4%	19.9
[20, 60, 20]	42%	53%	5%	19.9
[20, 20, 60]	41%	55%	4%	20
[40, 40, 20]	43%	53%	4%	19.9
[40, 20, 40]	41%	54%	5%	19.9
[20, 40, 40]	42%	54%	4%	20

with the $[0, 0, 100]$ weight configuration whose improvement percentage was the lowest.

Table A.17: Results of ICF for dataset 1.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	19%	18%	63%	1.3
[100, 0, 0]	7%	10%	83%	1.3
[0, 100, 0]	32%	29%	39%	1.2
[0, 0, 100]	32%	32%	36%	1.3
[60, 20, 20]	12%	14%	74%	1.3
[20, 60, 20]	24%	22%	54%	1.3
[20, 20, 60]	26%	25%	49%	1.3
[40, 40, 20]	18%	17%	65%	1.3
[40, 20, 40]	20%	19%	61%	1.3
[20, 40, 40]	22%	24%	54%	1.2

Table A.17 shows the results of the evaluation of the proposed extension of ICF, a sub-technique of ST-MVL, on dataset 1. The highest improvement percentage achieved was 32% with both weight configurations $[0, 100, 0]$ and $[0, 0, 100]$.

The high percentage of the measurements in this experiment remained unchanged with our extensions for this sub-technique. The unchanged measurements percentage reached 83% for the weight configuration $[100, 0, 0]$.

Table A.18 shows the results of the evaluation of the proposed extension of

Table A.18: Results of ICF for dataset 2.

Weights%	Improved	Worsened	Unchanged	RMSE
[33, 33, 34]	22%	25%	53%	28.3
[100, 0, 0]	14%	23%	63%	28.3
[0, 100, 0]	26%	25%	49%	28.3
[0, 0, 100]	34%	35%	31%	28.3
[60, 20, 20]	18%	21%	61%	28.3
[20, 60, 20]	22%	22%	56%	28.3
[20, 20, 60]	28%	21%	41%	28.3
[40, 40, 20]	19%	22%	58%	28.3
[40, 20, 40]	23%	27%	50%	28.3
[20, 40, 40]	25%	25%	50%	28.3

ICF, a sub-technique of ST-MVL, on dataset 2. The improved measurements percentages varied slightly for dataset 2 from that datasest 1 with an insignificant improvement for this setting. The unchanged measurements percentages remain relatively high for almost more than 50% with all the weight configurations.

Summary of the results of the extensions of the sub-techniques of ST-MVL
Our extension not having an impact on IDW for dataset 2 may indicate that even though the technique gives better results, in this case when neighboring sensors are ensured to exist in close proximity and similar indoor environment, which is dataset 1, our extensions can still show significant percentages of improvement.

SES technique relies on time-neighboring measurements from the same sensor, so enriching this technique with sensor quality was inapplicable as the data comes from the same sensor with the missing measurement.

Unlike the other studied techniques, UCF performed very poorly on dataset 1 and slightly better with dataset 2. With ICF extension, we notice higher percentages of unchanged values for both data settings.

B - My Publications

[1] Mehanna, S., Berchiche, N., Niare, A. B., Maaradji, A., Hacid, H., and Soukane, A. (2023). Towards Real-Time Image Mining for Waste Detection in Deserts. In 2023 Innovation and Technological Advances for Sustainable Development (ITAS), Doha, Qatar.

[2] Mehanna,, S., Maaradji, A., Kedad, Z., Chachoua, M., Soukane, A., and Hacid, H. (2023). Towards Improving Data Completeness in Drone-Based Waste Detection. In The 7th Conference for Information & Communication Technologies for Organization & Society (ICTO), Paris, France.

[3] Mehanna, S., Kedad, Z., and Chachoua, M. (2022). Completeness assessment and improvement in mobile crowd-sensing environments. SN Comput. Sci., 3(3):216.

[4] Brahem, M., Hafyani, H. E., Mehanna, S., Zeitouni, K., Yeh, L., Taher, Y., Kedad, Z., Ktaish, A., Chachoua, M., and Ray, C. (2021). 12 - data perspective on environmental mobile crowd sensing. In Bhattacharyya, S., Mondal, N. K., Platos, J., Snášel, V., and Krömer, P., editors, Intelligent Environmental Data Monitoring for Pollution Management, Intelligent Data-Centric Systems, pages 269–288 Academic Press.

[5] Mehanna, S., Kedad, Z., and Chachoua, M. (2020). Com- pleteness issues in mobile crowd-sensing environments. In Marchiori, M., Mayo, F. J. D., and Filipe, J., editors, Proceedings of the 16th International Conference on Web Information Systems and Technologies, WEBIST 2020, Budapest, Hungary, November 3-5, 2020, pages 129–138. SCITEPRESS: Winner of the best student paper award.

[6] Brahem M., Chachoua M., El Hafyani H., Kedad Z., Ktaish A., Mehanna S., Ray C., Taher Y., Thibaud R., Yeh L., Zeitouni K. Polluscope - Vers un observatoire participatif de l'exposition individuelle à la pollution de l'air et de ses effets sanitaires. SAGEO 2019, Clermont-Ferrand 13-15/11/2019.

Bibliography

[Aalen, 1989] Aalen, O. O. (1989). A linear regression model for the analysis of life times. *Statistics in medicine*, 8(8):907–925.

[Abboud et al., 2021] Abboud, M., Hafyani, H. E., Zuo, J., Zeitouni, K., and Taher, Y. (2021). Micro-environment recognition in the context of environmental crowdsensing. In Costa, C. and Pitoura, E., editors, *Proceedings of the Workshops of the EDBT/ICDT 2021 Joint Conference, Nicosia, Cyprus, March 23, 2021*, volume 2841 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Aggarwal, 2016] Aggarwal, C. (2016). *Outlier Analysis*. Springer International Publishing.

[Alfred., 1972] Alfred., T. (1972). *Logique, sémantique, méta-mathématique*. Armand Colin.

[Alizamini et al., 2010] Alizamini, F. G., Pedram, M. M., Alishahi, M., and Badie, K. (2010). Data quality improvement using fuzzy association rules. *2010 International Conference on Electronics and Information Engineering*, 1:V1–468–V1–472.

[Alvear et al., 2018] Alvear, O., Calafate, C. T., Cano, J.-C., and Manzoni, P. (2018). Crowdsensing in smart cities: Overview, platforms, and environment sensing issues. *Sensors*, 18(2).

[Angiulli and Fassetti, 2007] Angiulli, F. and Fassetti, F. (2007). Detecting distance-based outliers in streams of data. In Silva, M. J., Laender, A. H. F., Baeza-Yates, R. A., McGuinness, D. L., Olstad, B., Olsen, Ø. H., and Falcão, A. O., editors, *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pages 811–820. ACM.

[Aquino et al., 2019] Aquino, G. R. C. D., de Farias, C. M., and Pirmez, L. (2019). Hygieia: data quality assessment for smart sensor network. In Hung, C. and Papadopoulos, G. A., editors, *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*, pages 889–891. ACM.

[Audibert et al., 2020] Audibert, J., Michiardi, P., Guyard, F., Marti, S., and Zuluaga, M. A. (2020). USAD: unsupervised anomaly detection on multivariate time series. In Gupta, R., Liu, Y., Tang, J., and Prakash, B. A., editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3395–3404. ACM.

[Azimi and Pahl, 2021] Azimi, S. and Pahl, C. (2021). The effect of iot data completeness and correctness on explainable machine learning models. In Strauss, C., Kotsis, G., Tjoa, A. M., and Khalil, I., editors, *Database and Expert Systems Applications - 32nd International Conference, DEXA 2021, Virtual Event, September 27-30, 2021, Proceedings, Part II*, volume 12924 of *Lecture Notes in Computer Science*, pages 151–160. Springer.

[Bakar et al., 2006] Bakar, Z. A., Mohemad, R., Ahmad, A., and Deris, M. M. (2006). A comparative study for outlier detection techniques in data mining. In *2006 IEEE Conference on Cybernetics and Intelligent Systems*, pages 1–6.

[Ballou et al., 1998] Ballou, D., Wang, R., Pazer, H., and Tayi, G. K. (1998). Modeling information manufacturing systems to determine information product quality. *Management Science*, 44(4):462–484.

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

[Batini and Scannapieco, 2016] Batini, C. and Scannapieco, M. (2016). *Data and Information Quality - Dimensions, Principles and Techniques*. Data-Centric Systems and Applications. Springer.

[Belussi et al., 2018] Belussi, A., Migliorini, S., and Eldawy, A. (2018). Detecting skewness of big spatial data in SpatialHadoop. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, pages 432–435.

[Bernd et al., 2007] Bernd, H., Marcus, K., Marcus, and Mathias (2007). How to measure data quality? - a metric based approach.

[Berrahou et al., 2015] Berrahou, L., Lalande, N., Serrano, E., Molla, G., Berti-Équille, L., Bimonte, S., Bringay, S., Cernesson, F., Grac, C., Ienco, D., Ber, F. L., and Teisseire, M. (2015). A quality-aware spatial data warehouse for querying hydroecological data. *Comput. Geosci.*, 85:126–135.

[Berti-Équille et al., 2011] Berti-Équille, L., Comyn-Wattiau, I., Cosquer, M., Kedad, Z., Nugier, S., Peralta, V., Cherfi, S. S., and Thion-Goasdoué, V. (2011). Assessment and analysis of information quality: a multidimensional model and case studies. *Int. J. Inf. Qual.*, 2(4):300–323.

[Bezdek et al., 1984] Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences*, 10(2-3):191–203.

[Bimonte et al., 2006] Bimonte, S., Tchounikine, A., and Miquel, M. (2006). Geocube, a multidimensional model and navigation operators handling complex measures: Application in spatial OLAP. In Yakhno, T. M. and Neuhold,

E. J., editors, *Advances in Information Systems, 4th International Conference, ADVIS 2006, Izmir, Turkey, October 18-20, 2006, Proceedings*, volume 4243 of *Lecture Notes in Computer Science*, pages 100–109. Springer.

[Bishop, 2016] Bishop, C. (2016). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer New York.

[Biswas et al., 2006] Biswas, J., Naumann, F., and Qiu, Q. (2006). Assessing the completeness of sensor data. In Lee, M., Tan, K., and Wuwongse, V., editors, *Database Systems for Advanced Applications, 11th International Conference, DASFAA 2006, Singapore, April 12-15, 2006, Proceedings*, volume 3882 of *Lecture Notes in Computer Science*, pages 717–732. Springer.

[Blázquez-García et al., 2021] Blázquez-García, A., Conde, A., Mori, U., and Lozano, J. A. (2021). A review on outlier/anomaly detection in time series data. *ACM Comput. Surv.*, 54(3):56:1–56:33.

[Boulil et al., 2013] Boulil, K., Pinet, F., Bimonte, S., Carluer, N., Lauvernet, C., Cheviron, B., Miralles, A., and Chanet, J. (2013). Guaranteeing the quality of multidimensional analysis in data warehouses of simulation results: Application to pesticide transfer data produced by the MACRO model. *Ecol. Informatics*, 16:41–52.

[Bovee et al., 2003] Bovee, M., Srivastava, R. P., and Mak, B. (2003). A conceptual framework and belief-function approach to assessing overall information quality. *International journal of intelligent systems*, 18(1):51–74.

[Braei and Wagner, 2020] Braei, M. and Wagner, S. (2020). Anomaly detection in univariate time-series: A survey on the state-of-the-art. *CoRR*, abs/2004.00433.

[Brahem et al., 2021] Brahem, M., Hafyani, H. E., Mehanna, S., Zeitouni, K., Yeh, L., Taher, Y., Kedad, Z., Ktaish, A., Chachoua, M., and Ray, C. (2021). 12 - data perspective on environmental mobile crowd sensing. In Bhattacharyya, S., Mondal, N. K., Platos, J., Snášel, V., and Krömer, P., editors, *Intelligent Environmental Data Monitoring for Pollution Management*, Intelligent Data-Centric Systems, pages 269–288. Academic Press.

[Breunig et al., 2000a] Breunig, M. M., Kriegel, H., Ng, R. T., and Sander, J. (2000a). LOF: identifying density-based local outliers. In Chen, W., Naughton, J. F., and Bernstein, P. A., editors, *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, pages 93–104. ACM.

[Breunig et al., 2000b] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000b). Lof: identifying density-based local outliers. In *SIGMOD '00*.

[Brownrigg, 1984] Brownrigg, D. R. (1984). The weighted median filter. *Communications of the ACM*, 27(8):807–818.

[Bédard et al., 2006] Bédard, Y., Rivest, S., and Proulx, M.-J. (2006). *Spatial Online Analytical Processing (SOLAP)*.

[Cao et al., 2014] Cao, L., Yang, D., Wang, Q., Yu, Y., Wang, J., and Rundensteiner, E. A. (2014). Scalable distance-based outlier detection over high-volume data streams. In Cruz, I. F., Ferrari, E., Tao, Y., Bertino, E., and Trajcevski, G., editors, *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, pages 76–87. IEEE Computer Society.

[Carreño et al., 2019] Carreño, A., Inza, I., and Lozano, J. A. (2019). Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. *Artificial Intelligence Review*, 53:3575–3594.

[Chandola et al., 2009] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58.

[Chauhan and Vig, 2015] Chauhan, S. and Vig, L. (2015). Anomaly detection in ECG time signals via deep long short-term memory networks. In *2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015, Campus des Cordeliers, Paris, France, October 19-21, 2015*, pages 1–7. IEEE.

[Chu and Funderlic, 2002] Chu, M. T. and Funderlic, R. (2002). The centroid decomposition: Relationships between discrete variational decompositions and svds. *SIAM J. Matrix Anal. Appl.*, 23(4):1025–1044.

[Damiani and Spaccapietra, 2006] Damiani, M. L. and Spaccapietra, S. (2006). Spatial data warehouse modelling. *Processing and Managing Complex Data for Decision Support*.

[Dangeti, 2017] Dangeti, P. (July 2017). *Statistics for Machine Learning*. Packt Publishing.

[Dasu et al., 2016] Dasu, T., Duan, R., and Srivastava, D. (2016). Data Quality for Temporal Streams. Technical report.

[De Feo, 2017] De Feo, J. A. (2017). *Juran's quality handbook: The complete guide to performance excellence*. McGraw-Hill Education.

[Dessimond et al., 2021] Dessimond, B., Annesi-Maesano, I., Pepin, J.-L., Srairi, S., and Pau, G. (2021). Academically produced air pollution sensors for personal exposure assessment: The canarin project. *Sensors*, 21(5).

[Devillers et al., 2005] Devillers, R., Bédard, Y., and Jeansoulin, R. (2005). Multidimensional management of geospatial data quality information for its dynamic use within gis. *Photogrammetric Engineering & Remote Sensing*, 71(2):205–215.

[Devillers et al., 2007a] Devillers, R., Bédard, Y., Jeansoulin, R., and Moulin, B. (2007a). Towards spatial data quality information analysis tools for experts assessing the fitness for use of spatial data. *Int. J. Geogr. Inf. Sci.*, 21(3):261–282.

[Devillers et al., 2007b] Devillers, R., Bédard, Y., Jeansoulin, R., and Moulin, B. (2007b). Towards spatial data quality information analysis tools for experts assessing the fitness for use of spatial data. *International Journal of Geographical Information Science*, 21(3):261–282.

[Donoghue et al., 2011] Donoghue, J. O., Kane, T. O., Gallagher, J., Courtney, G., Aftab, A., Casey, A., Torres, J., and Angove, P. (2011). Modified early warning scorecard: the role of data/information quality within the decision making process. *Electronic Journal of Information Systems Evaluation*, 14(1):pp100–109.

[Ehrlinger and Wöß, 2018] Ehrlinger, L. and Wöß, W. (2018). A novel data quality metric for minimality. In Hacid, H., Sheng, Q. Z., Yoshida, T., Sarkheyli, A., and Zhou, R., editors, *Data Quality and Trust in Big Data - 5th International Workshop, QUAT 2018, Held in Conjunction with WISE 2018, Dubai, UAE, November 12-15, 2018, Revised Selected Papers*, volume 11235 of *Lecture Notes in Computer Science*, pages 1–15. Springer.

[Ehrlinger and Wöß, 2022] Ehrlinger, L. and Wöß, W. (2022). A survey of data quality measurement and monitoring tools. *Frontiers Big Data*, 5:850611.

[Emran, 2015] Emran, N. A. (2015). Data completeness measures. In Abraham, A., Muda, A. K., and Choo, Y.-H., editors, *Pattern Analysis, Intelligent Security and the Internet of Things*, pages 117–130, Cham. Springer International Publishing.

[Ergun et al., 2021] Ergun, K., Ayoub, R., Mercati, P., Liu, D., and Rosing, T. (2021). Energy and qos-aware dynamic reliability management of iot edge computing systems. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, ASPDAC '21, page 561–567, New York, NY, USA. Association for Computing Machinery.

[Evennou, 2007] Evennou, F. (2007). Advanced techniques and technologies for indoor mobile positioning.

[Fishbain et al., 2017] Fishbain, B., Lerner, U., Castell, N., Cole-Hunter, T., Popoola, O., Broday, D. M., Iñiguez, T. M., Nieuwenhuijsen, M., Jovasevic-Stojanovic, M., Topalovic, D., Jones, R. L., Galea, K. S., Etzion, Y., Kizel, F., Golumbic, Y. N., Baram-Tsabari, A., Yacobi, T., Drahler, D., Robinson, J. A., Kocman, D., Horvat, M., Svecova, V., Arpaci, A., and Bartonova, A. (2017). An evaluation tool kit of air quality micro-sensing units. *Science of the Total Environment*, 575:639–648.

[Fizza et al., 2022] Fizza, K., Jayaraman, P. P., Banerjee, A., Georgakopoulos, D., and Ranjan, R. (2022). Evaluating sensor data quality in internet of things smart agriculture applications. *IEEE Micro*, 42(1):51–60.

[Fox, 1972] Fox, A. J. (1972). Outliers in time series. *Journal of the royal statistical society series b-methodological*, 34:350–363.

[Fox et al., 1994] Fox, C. J., Levitin, A., and Redman, T. C. (1994). The notion of data and its quality dimensions. *Inf. Process. Manag.*, 30(1):9–20.

[Giannoni et al., 2018] Giannoni, F., Mancini, M., and Marinelli, F. (2018). Anomaly detection models for iot time series data. *CoRR*, abs/1812.00890.

[Goodfellow et al., 2014] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680.

[Grubbs, 1950] Grubbs, F. (1950). Sample criteria for testing outlying observations. In *Ann. Matiz. Statist.*

[Grubbs, 1969] Grubbs, F. (1969). Procedures for detecting outlying observations in samples. In *Technometrics*.

[Gupta et al., 2014] Gupta, M., Gao, J., Aggarwal, C., and Han, J. (2014). Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 26.

[Hackl and Ledolter, 1991] Hackl, P. and Ledolter, J. (1991). A control chart based on ranks. *Journal of Quality Technology*, 23:117–124.

[Han et al., 2010] Han, Q., Hakkarinen, D., Boonma, P., and Suzuki, J. (2010). Quality-aware sensor data collection. *International Journal of Sensor Networks*, 7(3):127.

[Hassany Shariat Panahy et al., 2013] Hassany Shariat Panahy, P., Sidi, F., Affendey, L., A. Jabar, M., Ibrahim, H., and Mustapha, A. (2013). Discovering

dependencies among data quality dimensions: A validation of instrument. *Journal of Applied Sciences*, 13:95–102.

[Hawkins, 1980] Hawkins, D. M. (1980). *Identification of outliers*, volume 11. Springer.

[He et al., 2002] He, Z., Xu, X., and Deng, S. (2002). Squeezer: An efficient algorithm for clustering categorical data. *J. Comput. Sci. Technol.*, 17(5):611–624.

[He et al., 2003] He, Z., Xu, X., and Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognit. Lett.*, 24(9-10):1641–1650.

[Heinrich and Klier, 2009] Heinrich, B. and Klier, M. (2009). A novel data quality metric for timeliness considering supplemental data. In Newell, S., Whitley, E. A., Pouloudi, N., Wareham, J., and Mathiassen, L., editors, *17th European Conference on Information Systems, ECIS 2009, Verona, Italy, 2009*, pages 2651–2662.

[Hinrichs, 2002] Hinrichs, H. (2002). *Datenqualitätsmanagement in data warehouse-systemen*. PhD thesis, Universität Oldenburg.

[Hochenbaum et al., 2017] Hochenbaum, J., Vallis, O. S., and Kejariwal, A. (2017). Automatic anomaly detection in the cloud via statistical learning. *CoRR*, abs/1704.07706.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

[Huang et al., 2022] Huang, L., Shen, S., Zhu, L., Shi, Q., and Zhang, J. (2022). Context-aware multi-scale aggregation network for congested crowd counting. *Sensors*, 22(9):3233.

[Idé, 2006] Idé, T. (2006). Why does subsequence time-series clustering produce sine waves? In Fürnkranz, J., Scheffer, T., and Spiliopoulou, M., editors, *Knowledge Discovery in Databases: PKDD 2006, 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Berlin, Germany, September 18-22, 2006, Proceedings*, volume 4213 of *Lecture Notes in Computer Science*, pages 211–222. Springer.

[iso25000,] iso25000. Iso ISO 25000 portal. `https://iso25000.com/index.php/en/iso-25000-standards/iso-25012/134-accuracy`. Accessed: 2023-01-24.

[Jacke et al., 2012] Jacke, C., Kalder, M., Wagner, U., and Albert, U.-S. (2012). Valid comparisons and decisions based on clinical registers and population based cohort studies: Assessing the accuracy, completeness and epidemiological relevance of a breast cancer query database. *BMC research notes*, 5:700.

[Jarke, 2003] Jarke, M. (2003). *Fundamentals of data warehouses, 2nd Edition*. Springer.

[Jesiļevska, 2017] Jesiļevska, S. (2017). Data quality dimensions to ensure optimal data quality. *Romanian Economic Journal*, 20(63).

[Joglekar and Kulkarni, 2016] Joglekar, P. and Kulkarni, V. (2016). Mobile crowd sensing for urban computing. *International Journal of Latest Trends in Engineering and Technology.*, 7(4):344–351.

[Jolliffe, 2011] Jolliffe, I. T. (2011). Principal component analysis. In Lovric, M., editor, *International Encyclopedia of Statistical Science*, pages 1094–1096. Springer.

[Juddoo et al., 2018] Juddoo, S., George, C., Duquenoy, P., and Windridge, D. (2018). Data governance in the health industry: Investigating data quality dimensions within a big data context. *Applied System Innovation*, 1(4):43.

[Kammoun, 2016] Kammoun, S. (2016). *Géolocalisation à l'intérieur d'un bâtiment pour terminaux mobiles*. PhD thesis.

[Karagulian et al., 2019] Karagulian, F., Barbiere, M., Kotsev, A., Spinelle, L., Gerboles, M., Lagler, F., Redon, N., Crunaire, S., and Borowiak, A. (2019). Review of the performance of low-cost sensors for air quality monitoring. *Atmosphere*.

[Khayati et al., 2014] Khayati, M., Böhlen, M. H., and Gamper, J. (2014). Memory-efficient centroid decomposition for long time series. In Cruz, I. F., Ferrari, E., Tao, Y., Bertino, E., and Trajcevski, G., editors, *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, pages 100–111. IEEE Computer Society.

[Khayati et al., 2020] Khayati, M., Lerner, A., Tymchenko, Z., and Cudré-Mauroux, P. (2020). Mind the gap: An experimental evaluation of imputation of missing values techniques in time series. *Proc. VLDB Endow.*, 13(5):768–782.

[Kim et al., 2015] Kim, H., Choo, J., Kim, J., Reddy, C. K., and Park, H. (2015). Simultaneous discovery of common and discriminative topics via joint nonnegative matrix factorization. In Cao, L., Zhang, C., Joachims, T., Webb, G. I., Margineantu, D. D., and Williams, G., editors, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 567–576. ACM.

[Klein et al., 2007] Klein, A., Do, H. H., Hackenbroich, G., Karnstedt, M., and Lehner, W. (2007). Representing data quality for streaming and static data. *Proceedings - International Conference on Data Engineering*, (January 2014):3–10.

[Knorr and Ng, 1998] Knorr, E. M. and Ng, R. T. (1998). Algorithms for mining distance-based outliers in large datasets. In Gupta, A., Shmueli, O., and Widom, J., editors, *VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases, August 24-27, 1998, New York City, New York, USA*, pages 392–403. Morgan Kaufmann.

[Kontaki et al., 2011] Kontaki, M., Gounaris, A., Papadopoulos, A. N., Tsichlas, K., and Manolopoulos, Y. (2011). Continuous monitoring of distance-based outliers over data streams. In Abiteboul, S., Böhm, K., Koch, C., and Tan, K., editors, *Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11-16, 2011, Hannover, Germany*, pages 135–146. IEEE Computer Society.

[Koren et al., 2009] Koren, Y., Bell, R. M., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

[Köpcke et al., 2013] Köpcke, F., Trinczek, B., Majeed, R., Schreiweis, B., Wenk, J., Leusch, T., Ganslandt, T., Ohmann, C., Bergh, B., Röhrig, R., Dugas, M., and Prokosch, H.-U. (2013). Evaluation of data completeness in the electronic health record for the purpose of patient recruitment into clinical trials: A retrospective analysis of element presence. *BMC medical informatics and decision making*, 13:37.

[Langefors and Sundgren, 1975] Langefors, B. and Sundgren, B. (1975). *Information systems architecture*. Petrocelli/Charter.

[Languille et al., 2020] Languille, B., Gros, V., Bonnaire, N., Pommier, C., Honoré, C., Debert, C., Gauvin, L., Srairi, S., Annesi-Maesano, I., Chaix, B., and Zeitouni, K. (2020). A methodology for the characterization of portable sensors for air quality measure with the goal of deployment in citizen science. *Science of The Total Environment*, 708:134698.

[Lee et al., 2009] Lee, Y. W., Pipino, L. L., Funk, J. D., and Wang, R. Y. (2009). Journey to data quality.

[Li et al., 2009] Li, L., McCann, J., Pollard, N. S., and Faloutsos, C. (2009). Dynammo: mining and summarization of coevolving sequences with missing values. In IV, J. F. E., Fogelman-Soulié, F., Flach, P. A., and Zaki, M. J., editors, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 507–516. ACM.

[Liang and Parthasarathy, 2016] Liang, J. and Parthasarathy, S. (2016). Robust contextual outlier detection: Where context meets sparsity. *CoRR*, abs/1607.08329.

[Liu et al., 2019] Liu, C., Nitschke, P., Williams, S., and Zowghi, D. (2019). Data quality and the internet of things. *Computing*.

[Liu et al., 2017] Liu, C., Talaei-Khoei, A., Zowghi, D., and Daniel, J. (2017). Data completeness in healthcare: A literature survey. *Pac. Asia J. Assoc. Inf. Syst.*, 9(2):5.

[Liu et al., 2016] Liu, Y., Li, J., and Zou, Z. (2016). Determining the real data completeness of a relational dataset. *J. Comput. Sci. Technol.*, 31(4):720–740.

[Liu et al., 2020] Liu, Y., Zhou, Y., and Lu, J. (2020). Exploring the relationship between air pollution and meteorological conditions in china under environmental governance. *Scientific Reports*, 10(14518).

[López et al., 2005] López, I. F. V., Snodgrass, R. T., and Moon, B. (2005). Spatiotemporal aggregate computation: a survey. *IEEE Trans. Knowl. Data Eng.*, 17(2):271–286.

[Ma et al., 2014] Ma, H., Zhao, D., and Yuan, P. (2014). Opportunities in mobile crowd sensing. *IEEE Communications Magazine*, 52:29–35.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations.

[Malhotra et al., 2015] Malhotra, P., Vig, L., Shroff, G. M., and Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. In *ESANN*.

[Mazumder et al., 2010] Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *J. Mach. Learn. Res.*, 11:2287–2322.

[McGilvray, 2021] McGilvray, D. (2021). *Executing data quality projects: Ten steps to quality data and trusted information (TM)*. Academic Press.

[Mehanna et al., 2020] Mehanna, S., Kedad, Z., and Chachoua, M. (2020). Completeness issues in mobile crowd-sensing environments. In Marchiori, M., Mayo, F. J. D., and Filipe, J., editors, *Proceedings of the 16th International Conference on Web Information Systems and Technologies, WEBIST 2020, Budapest, Hungary, November 3-5, 2020*, pages 129–138. SCITEPRESS.

[Nonnemacher et al., 2014] Nonnemacher, M., Nasseh, D., and Stausberg, J. (2014). *Datenqualität in der medizinischen Forschung: Leitlinie zum adaptiven Management von Datenqualität in Kohortenstudien und Registern*.

[Okafor et al., 2020] Okafor, N., Alghorani, Y., and Delaney, D. (2020). Towards improvement in data quality of low-cost iot sensors in environmental monitoring networks using data fusion and machine learning approach. *ICT Express*, 6.

[Östman, 1997] Östman, A. (1997). The specification and evaluation of spatial data quality. *Proceedings of the 18st International Cartographic Conference*, pages 836–847.

[Peralta, 2006] Peralta, V. (2006). *Data Quality Evaluation in Data Integration Systems*. PhD thesis, Versailles Saint-Quentin-en-Yvelines University, France.

[Pipino et al., 2002] Pipino, L., Lee, Y. W., and Wang, R. Y. (2002). Data quality assessment. *Commun. ACM*, 45:211–218.

[qin Han et al., 2011] qin Han, S., Bian, H., Feng, Y., Liu, A., jin Li, X., Zeng, F., and Zhang, X. (2011). Analysis of the relationship between o3, no and no2 in tianjin, china. *Aerosol and Air Quality Research*, 11:128–139.

[Ray, 2018] Ray, C. (2018). Data variety and integrity assessment for maritime anomaly detection. *CEUR Workshop Proceedings*, 2343:4–7.

[Redman, 2005] Redman, T. (2005). Measuring data accuracy: A framework and review. *Information Quality*, pages 21–36.

[Rodríguez and Servigne, 2013] Rodríguez, C. C. G. and Servigne, S. (2013). Managing Sensor Data Uncertainty. *International Journal of Agricultural and Environmental Information Systems*, 4(1):35–54.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation.

[Safaei et al., 2020] Safaei, M., Asadi, S., Driss, M., Boulila, W., Alsaeedi, A., Chizari, H., Abdullah, R., and Safaei, M. (2020). A systematic literature review on outlier detection in wireless sensor networks. *Symmetry*, 12.

[Segev and Fang, 1989] Segev, A. and Fang, W. (1989). Currency-based updates to distributed materialized views.

[Serhani et al., 2016] Serhani, M. A., Kassabi, H. T. E., Taleb, I., and Nujum, A. R. (2016). An hybrid approach to quality evaluation across big data value chain. In Pu, C., Fox, G. C., and Damiani, E., editors, *2016 IEEE International Congress on Big Data, San Francisco, CA, USA, June 27 - July 2, 2016*, pages 418–425. IEEE Computer Society.

[Sidi et al., 2012] Sidi, F., Panah, P. H. S., Affendey, L. S., Jabar, M. A., Ibrahim, H., and Mustapha, A. (2012). Data quality: A survey of data quality dimensions. In Mahmod, R., Abdullah, R., Abdullah, L. N., Sembok, T. M. T., Smeaton, A. F., Crestani, F., Doraisamy, S., Kadir, R. A., and Ismail, M., editors, *2012 International Conference on Information Retrieval & Knowledge Management, Kuala Lumpur, Malaysia, March 13-15, 2012*, pages 300–304. IEEE.

[Skillicorn, 2007] Skillicorn, D. (2007). *Understanding Complex Datasets: Data Mining with Matrix Decompositions*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. CRC Press.

[Thomas and J.E, 2021] Thomas, R. and J.E, J. (2021). A novel ensemble method for detecting outliers in categorical data. *International Journal of Advanced Trends in Computer Science and Engineering*, 9:4947–4953.

[Tobler, 1970] Tobler, W. (1970). A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46:234–240.

[Todoran et al., 2015] Todoran, I.-G., Lecornu, L., Khenchaf, A., and Caillec, J.-M. L. (2015). A methodology to evaluate important dimensions of information quality in systems. *Journal of Data and Information Quality (JDIQ)*, 6(2-3):1–23.

[Tran et al., 2016] Tran, L. V., Fan, L., and Shahabi, C. (2016). Distance-based outlier detection in data streams. *Proc. VLDB Endow.*, 9(12):1089–1100.

[Troyanskaya et al., 2001] Troyanskaya, O. G., Cantor, M. N., Sherlock, G., Brown, P. O., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinform.*, 17(6):520–525.

[Tsay, 1988] Tsay, R. S. (1988). Outliers, level shifts, and variance changes in time series. *Journal of Forecasting*, 7:1–20.

[Tsay et al., 2000] Tsay, R. S., Peña, D., and Pankratz, A. E. (2000). Outliers in multivariate time series. *Biometrika*, 87(4):789–804.

[Wang et al., 2016] Wang, L., Zhang, D., Wang, Y., Chen, C., Han, X., and M'Hamed, A. (2016). Sparse mobile crowdsensing: Challenges and opportunities. *IEEE Communications Magazine*, 54(7):161–167.

[Wang and Strong, 1996] Wang, R. Y. and Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *J. Manag. Inf. Syst.*, 12(4):5–33.

[Xiao et al., 2017] Xiao, Y., Bochner, A., Makunike, B., Holec, M., Xaba, S., Tshimanga, M., Chitimbire, V., Barnhart, S., and Feldacker, C. (2017). Challenges in data quality: the influence of data quality assessments on data availability and completeness in a voluntary medical male circumcision programme in zimbabwe. *BMJ open*, 7(1):e013562.

[Yang et al., 2009] Yang, D., Rundensteiner, E. A., and Ward, M. O. (2009). Neighbor-based pattern detection for windows over streaming data. In Kersten, M. L., Novikov, B., Teubner, J., Polutin, V., and Manegold, S., editors,

EDBT 2009, 12th International Conference on Extending Database Technology, Saint Petersburg, Russia, March 24-26, 2009, Proceedings, volume 360 of *ACM International Conference Proceeding Series*, pages 529–540. ACM.

[Yi et al., 2016] Yi, X., Zheng, Y., Zhang, J., and Li, T. (2016). ST-MVL: filling missing values in geo-sensory time series data. In Kambhampati, S., editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2704–2710. IJCAI/AAAI Press.

[Yoseph and Heikkilä, 2019] Yoseph, F. and Heikkilä, M. (2019). A clustering approach for outliers detection in a big point-of-sales database. In *2019 International Conference on Machine Learning and Data Engineering (iCMLDE)*, pages 65–71.

[Zappatore et al., 2019] Zappatore, M., Loglisci, C., Longo, A., Bochicchio, M., Vaira, L., and Malerba, D. (2019). Trustworthiness of context-aware urban pollution data in mobile crowd sensing. *IEEE Access*, PP:1–1.

[Zappatore et al., 2017] Zappatore, M., Longo, A., and Bochicchio, M. (2017). Crowd-sensing our smart cities: a platform for noise monitoring and acoustic urban planning. *Journal of Communications Software and Systems*, 13:53.

[Zhang et al., 2020] Zhang, M., Guo, J., Li, X., and Jin, R. (2020). Data-driven anomaly detection approach for time-series streaming data. *Sensors*, 20(19):5646.

[Zheng et al., 2017] Zheng, G., Brantley, S. L., Lauvaux, T., and Li, Z. (2017). Contextual spatial outlier detection with metric learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 2161–2170. ACM.