UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Rain Hallikas

Multivocal Literature Review on Data Quality Challenges in Data Pipelines

Master's Thesis (30 ECTS)

Supervisors:
Dietmar Alfred Paul Kurt Pfahl, PhD
Mario Ezequiel Scott, PhD

Tartu 2024

Multivocal Literature Review on Data Quality Challenges in Data Pipelines

Abstract:

This thesis presents a multivocal literature review focusing on data quality challenges within data pipelines. Data quality is intently affected by data pipeline processes, and this thesis aims to provide a nuanced understanding of most popular aspects to influence data quality within data pipelines. The multivocal nature of the thesis introduces grey literature into the research to have more precise conclusions on the topic, as data pipelines have only surged in popularity in recent years. Additionally, the thesis offers an overview of current solutions and open issues with data quality to advance data pipeline engineering. The challenges together with solutions represent a guide to understanding data quality challenges within pipelines more deeply and offer insight for future work.

Keywords:

Data pipelines, data quality, challenges, multivocal literature review

CERCS: P175 Informatics, systems theory

Multivokaalne kirjanduse ülevaade andmekvaliteedi probleemidest andmetorudes

Lühikokkuvõte:

See magistritöö annab süstemaatilise ülevaate andmekvaliteedi probleemidest ja väljakutsetest andmetorudes. Andmetorude protsessid töötlevad andmeid mitmel erineval moel ja magistritöö üritab leida enim mainitud probleemid, mis andmekvaliteeti andmetorudes mõjutavad. Töös on kasutatud multivokaalset lähenemist, mis käsitleb teaduslikele uuringutele lisaks ka halli kirjandust, et anda andmetorude vähesest käsitlusest tulenemata täpseid tulemusi. Töö annab lisaks ülevaate nii lahendustest kui ka avatud küsimustest, et andmetorude ehitamise valdkonda edasi arendada. Töös teostatud andmekvaliteedi probleemide kaardistus koos lahendustega on juhend, et andmekvaliteedi väljakutseid andmetorudes paremini mõista ja leida kohti edaspidisteks uuringuteks.

Võtmesõnad:

Andmetorud, andmekvaliteet, väljakutsed, probleemid, multivokaalne kirjanduse ülevaade

CERCS: P175 Informaatika, süsteemiteooria

Table of Contents

Introduction

In today's digital era, data pipelines are widely recognized as essential tools for managing and processing substantial amounts of data efficiently. They serve as the backbone of modern data systems, helping organizations collect, process, and utilize data from various sources effectively. These pipelines streamline the flow of data, making it easier for organizations to extract valuable insights.

The popularity of data pipelines has grown due to the increasing volume of data generated by businesses and the availability of cloud computing services in the early 2010s. A significant catalyst for this trend occurred in 2012 with the release of AWS Data Pipeline, facilitating seamless data movement and processing across various AWS compute and storage services [1]. From that point on, the volume of results related to data pipelines on Google Scholar has consistently increased until 2021 and have maintained decent popularity since. However, academic focus on maintaining data quality within data pipelines is still limited, with only Foidl et al. [2] offering a comprehensive analysis on root causes of data-related issues in another multivocal literature review in the realm of data pipelines.

Finding good practices for designing data pipelines can be a broad subject ranging from purely data-related challenges to more general organizational issues. The aim of the thesis is to find mentions of common issues influencing data quality within data pipelines and expand on their solutions and current stature to guide future work. The thesis finds contributions from empirical studies, and grey literature to add depth to the research.

The thesis finds answers to the following research questions (RQs) -

RQ1: What challenges exist in ensuring data quality throughout the various stages of data pipelines?

RQ2: What are the current solutions or mitigation strategies to address data quality challenges in data pipelines?

RQ3: What are the open issues related to data pipelines that ensure data quality?

The first chapter of the thesis discusses the background of data pipelines, data quality, and their interconnectedness. The second chapter gathers related academic research on this topic. The third chapter gives an overview of the methodology used in this multivocal literature review. The fourth chapter gathers the results and answers the research questions. The fifth chapter discusses the results and offers an overview of the results. The sixth chapter concludes the entirety of the thesis.

1 Background

This chapter focuses on the fundamental concepts of data pipelines, providing definitions for the context of the thesis and emphasizing the role of data quality within data pipelines.

1.1 Unpacking Data Pipelines

At the core of data engineering, data pipelines serve as the backbone of many modern data-driven operations in organizations [2]. Data pipelines streamline the collection, transformation, and delivery of data, making them indispensable for effective data utilization. Data quality is one of the cornerstones in this context, as its aspects are thoroughly involved in all data pipeline stages. Maintaining data quality throughout the data pipeline is critical as it has a significant impact on the output of the pipeline and therefore also a considerable impact on the performance of data-driven operations and applications.

Data pipelines can operate in diverse ways, depending on the operation or need. Batch processing pipelines operate with a dataset (e.g., monthly payroll and billing systems). Real-time data pipelines operate in as short a period as possible (e.g., purchasing a stock within milliseconds of receiving payment). [3]

Data pipeline process starts with sources and data ingestion. After the data is extracted, it is manipulated according to requirements. This data manipulation often includes steps like transformation, augmentation, filtering, grouping, and aggregation. Finally, the processed data arrives at a data lake or data warehouse for analysis. [4]

Munappy et al. [5] categorize challenges with data pipelines into three larger categories: infrastructural, organizational and data quality related. This thesis narrows its focus to solely data quality challenges when building data pipelines.

1.2 Defining Data Pipelines

The concept of data pipeline is defined based on trusted sources such as IBM, AWS Amazon, and Snowflake. The thesis will examine their data pipeline definitions:

A data pipeline is a method in which raw data is ingested from various data sources and then ported to a data store, like a data lake or data warehouse, for analysis. - **IBM** [6]

A data pipeline is a series of processing steps to prepare enterprise data for analysis. - **AWS Amazon** [7]

A data pipeline is a means of moving data from one place (the source) to a destination (such as a data warehouse). Along the way, data is transformed and optimized, arriving in a state that can be analysed and used to develop business insights. - **Snowflake** [4]

Therefore, from these definitions the key characteristics of a data pipeline are as follows:

1. Source data is raw, unprocessed, and considered unoptimized.
2. Source data is scattered in different forms in various data sources.
3. Source data needs to be ingested for the data pipeline.
4. Data pipeline tries to process, transform, and optimize the ingested data.
5. Data pipeline outputs the data in a state that can be analysed and used for a purpose.
6. The output data is ported to a data store (data lake or data warehouse).

To ensure a precise understanding of data pipelines, the author proposes a new and improved detailed definition of a data pipeline:

Data pipeline is a process where raw, unoptimized data is ingested from diverse data sources for transformation and then ported to a data store in an optimized state where it is ready for analysis.

This definition aligns most closely with the IBM definition with the included mentioning of data transformation and optimization. Throughout this thesis, the proposed definition is aimed to provide a clear and consistent framework for gathering information from diverse sources, facilitating a cohesive analysis and synthesis of data-quality related insights.

The proposed definition also aligns with ETL process in data pipelines. ETL stands for extract, transform, and load, offering a clear distinction for the main pipeline activities [8].

Figure 1 shows the flow of an ETL pipeline, which corresponds to the proposed definition.

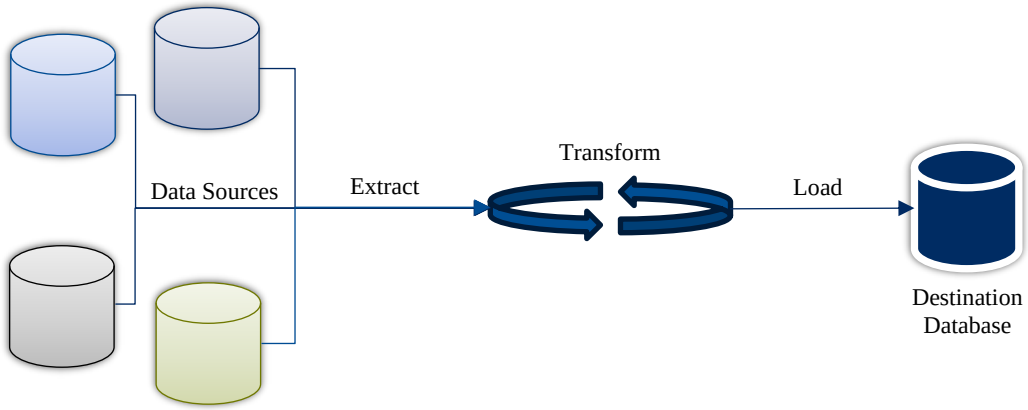

Figure 1 ETL Pipeline

1.3 Data Quality in Data Pipelines

It is essential to establish a clear understanding of the concept of data quality within the specific context of building data pipelines. The thesis is not trying to explore data quality challenges by themselves, but rather to focus on the challenges presented when integrating data quality considerations into the pipeline development process throughout its stages.

IBM defines data quality as follows:

Data quality measures how well a dataset meets criteria for accuracy, completeness, validity, consistency, uniqueness, timeliness, and fitness for purpose, and it is critical to all data governance initiatives within an organization. [9]

In the context of data pipelines, data quality extends to monitoring and preserving data quality as it traverses through various stages. All the data quality criteria need to be monitored throughout the stages of ingestion, transformation, and data store transportation. The key characteristics previously extracted from different data pipeline definitions are integral in the quality criteria monitoring process.

2 Related Work

Currently, there is a noticeable absence of systematic literature reviews specifically addressing data quality issues within data pipelines. However, [2] stands out as the lone multivocal literature review that finds deep-root causes for data-related issues in data pipelines. The main limitation of [2] is its approach of only handling bad data practices, and it does not focus on elements within data pipelines that indirectly influence data quality. While numerous studies have delved into data quality challenges, as indicated in [2], there are not many data quality studies within the context of data pipelines.

Rashid and Torchiano [10] conducted a systematic literature review on open data quality, extensively categorizing internal and external quality challenges. Their findings included a range of data quality challenges with metadata, data structure, data publishing, and data access. Oliveira et al. [11], in conjunction with similar results from [12], offer an extensive list of data quality challenges. These challenges are categorized into four levels: attribute/tuple, single relation, multiple relations, and multiple data sources. Although they provide concrete examples of potential data quality challenges within data pipelines (i.e. multiple data source challenges), the interconnections are not made. While [10], [11] and [12] all offer comprehensive insights into data quality challenges, they do not specifically address data pipelines.

In a multi-case study, Munappy et al. [5] come closest to defining concrete challenges within the context of data pipelines. In terms of data quality, they identify missing data files, operational errors, and logical changes as key challenges and present general solutions that include not only data quality challenges but also infrastructural and operational challenges. Munappy et al. [13] also tackle general data pipelines challenges and provide data pipeline modelling principles to solve those issues. While both [5] and [13] focus on challenges within data pipelines, they lack thorough coverage of data quality issues.

Foidl et al. [2] focus on identifying the root causes of data-related issues in data pipelines. They identify influencing factors for data-related issues, including the data itself, development and deployment, infrastructure, processing, and life cycle management. The root causes of data-related issues within the pipeline range from misplacement of symbols and characters to issues with raw data, lack of functions, problematic data frames, large input dataset sizes, and logical errors.

In summary, while [10], [11], and [12] provide extensive insights into data quality challenges, they do not specifically address data pipelines. Both [5] and [13] focus on data pipeline challenges, including some aspects of data quality, but do not provide comprehensive coverage of data quality issues. [2], on the other hand, stands out as a comprehensive analysis of data quality influencing factors within data pipelines, offering valuable insights into this domain.

3 Methodology

This thesis is a multivocal literature review that follows guidelines set by Garousi et al. [14]. Multivocal literature review is a form of Systematic Literature Review (SLR) which includes grey literature (e.g., blog posts and videos) in addition to formal literature [14].

Data pipeline architecture has seen a lot of development in recent years but the in-depth formal literature for specific aspects of data pipelines seemed limited based on the preliminary searches conducted on Ebsco, Google Scholar, and Google. The thesis aims to solve the literature scarcity problem via grey literature to provide a more comprehensive analysis.

3.1 Procedure

This section defines research questions, search process, filtering, data extraction, and finally the data synthesis strategy. Figure 2 presents the full procedure and flow of the thesis. Preliminary investigative search on Ebsco, Google Scholar and Google assists in defining research questions and search strings. Cut-offs are applied to the initial results from the final search. The final selection of sources is acquired after applying exclusion, inclusion, and quality criteria. These sources are fully read, and the data is extracted and classified for answering the research questions. The procedure is explained more deeply in the following chapters.

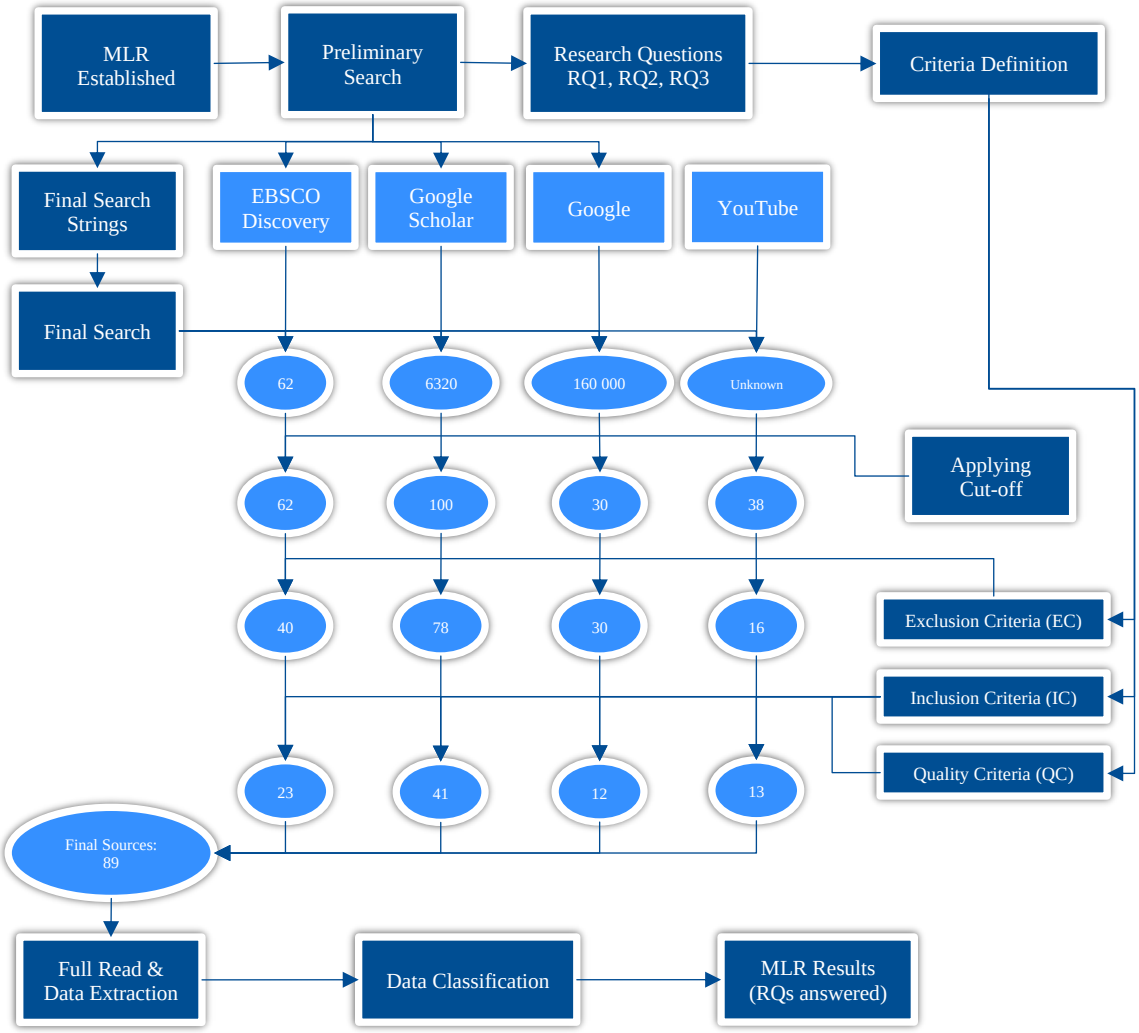

Figure 2 Research Procedure

3.1.1 Research Questions

The thesis aims to find sources related to data quality challenges in building data pipelines. The research questions are set based on the aim and justified as follows:

RQ1: What challenges exist in ensuring high data quality throughout the various stages of data pipelines?

Gathering challenges related to data quality in data pipelines is the cornerstone of the thesis. The specification of associating the challenge with data pipeline stage helps to pinpoint the challenges and make them more distinguishable and exact.

RQ2: What are the current solutions or mitigation strategies to address data quality challenges in data pipelines?

The solutions and mitigation strategies are directly aimed at the challenges found in RQ1. This provides an aspect to not only find the key problems with maintaining data quality throughout the data pipeline stages but also find viable solutions to them.

RQ3: What are the open issues related to data pipelines that ensure high data quality?

Identifying and exploring the challenges without proper solutions or mitigation strategies can give ground to future data pipeline research.

3.1.2 Search Process

The main search was conducted using EBSCO Discovery Service and Google Scholar. Additionally, grey literature from YouTube and Google were also included to gain additional insights as preliminary search and reading suggested that the subject might need extra insight. All sources use different search strings to find the most relevant results.

EBSCO. The primary source for this thesis was EBSCO Discovery Service [15]. The source produces accurate results as it has a sophisticated metadata search while maintaining a broad reach with the inclusion of major content providers like IEEE Xplore and ACM and hundreds more research databases [16].

EBSCO search string:

("data pipeline" OR "ETL pipeline" OR "data processing pipeline" OR "data engineering pipeline" OR "data flow pipeline" OR "data science pipeline") AND

(challenge OR problem OR issue OR difficulties OR difficulty OR error OR mitigation OR fault) AND

("data quality" OR "quality of data" OR "data validation" OR "data integrity")

The keywords were chosen to include three main ideas of the thesis: data pipeline, challenges, and data quality. The initial search included only these three terms and was expanded on multiple times. Different specific data pipelines were included to have more reach as data pipeline is a more recent and general term. This strategy was also used by Foidl et al. [2]. A similar approach was used for finding challenges and associating data quality with the search. The source type filters were not used to limit the search for only academic sources. The final search string provided an accurate yet comprehensive set of results for all the sources to be read and considered.

Google Scholar. The secondary source chosen was Google Scholar. Since Google Scholar returned a lot of results with the final EBSCO search string and introduced increasingly irrelevant sources in the latter pages of the result, the search string was modified. The search

string was stripped down to be broad and therefore also returned a more accurate and comprehensible list of sources.

Google Scholar search string:

("data pipeline" OR “data pipelines”) AND

(challenge OR problem OR issue OR difficulties OR difficulty OR error OR mitigation OR fault)

AND ("data quality" OR "quality of data" OR "data validation")

The specific pipelines were removed as they clouded the accuracy of the results. The term data validation was also removed for the same reason. The final search string provided a good number of duplicates compared to EBSCO results, which was a good sign.

YouTube.

The third source was introduced to compliment the result pool as data quality aspects were not as deeply covered in academic sources about data pipelines. The search was conducted without cookies to avoid YouTube algorithms suggesting anything other than the search terms.

YouTube search string:

"data pipeline" ("data quality" OR challenges OR issues OR problems)

A more general search string was chosen, as there was no inherent need to find a high number of alternative media sources. This approach helped with acquiring more accurate results as well.

Google.

The fourth approach in finding complimentary sources was to simply use Google search engine. Google would find news articles, blog posts, podcasts etc. that Google Scholar would not find, and therefore enable deeper insight into data quality issues within data pipelines.

Google search string:

"data pipeline" AND "data quality" AND (challenges OR issues OR problems)

Many different search strings were tried and assessed to avoid going through substantial amounts of inaccurate grey literature. As with YouTube, a more general approach was chosen as it helped find more relevant results.

3.1.3 Filtering

The selection criteria include inclusion, exclusion, and quality criteria. The criteria consider that the thesis will also include grey literature sources and the sources do not need to be academic or extensive. The quality criteria are especially targeted at grey literature to keep only trustworthy sources.

Exclusion criteria:

EC1: Sources not in English.

EC2: Sources not closely related to the data engineering domain or not in the computer science field.[1]

EC3: Sources for which full text or content is not found or available.

EC4: Duplicate sources.

EC5: Anonymous authors.

EC6: Sources published in 2012 or before.[2]

EC7: Secondary studies (SLR/MLR).

Inclusion criteria:

IC1: Sources addressing challenges related to data quality within the context of data pipelines.

IC2: Sources discussing solutions or mitigation strategies for data quality challenges within the context of data pipelines.

IC3: Sources proposing frameworks or models for addressing data quality challenges in data pipelines.

IC4: Sources providing analyses of different approaches to data quality in data pipelines, providing insight into effectiveness of various strategies.

Quality criteria:

QC1: The main goal and the results of the source are clearly stated.

QC2: The source is objective, without excess subjective opinions.

QC3: The source has a clear aim and structure.

QC4: The author of the source has affiliation or work related to data engineering domain.

3.1.4 Data Extraction

Table 1 presents an outline for the data extraction table. The identification data for each source entry includes the title, authors, year, and content provider. EBSCO sources also include bibliography type and subject. URL is also included for quick access. Each source entry has challenges, solutions and open issues included, associated with the research questions. The research question columns are broken down to specific data pipeline stages for a more concise data extraction.

Table 1 Data Extraction Table

Data Item	**Value**	**Relevance**
Title	Title of the source	Metadata
Author(s)	Names of the authors	
Year	Year when the source was published	

[1] Sources from other domains that are closely related to data engineering are not excluded.

[2] Sources before 2012 were considered inaccurate based on the preliminary searches.

Content Provider	Content provider of the source	
Bibliography Type	Bibliography type of the source (e.g., Conference Paper)	
Subject	Domain or field of the source	
URL	URL of the source for quick access	
Challenges	Challenges with data quality within different data pipeline stages	RQ1
Solutions	Solutions to the data quality challenges within data pipelines	RQ2
Open Issues	Specifically mentioned open issues related to data quality in data pipelines for future research	RQ3

The data extraction table is provided with the thesis and available with draft notes in [17]. The table provides the initial list of sources, list of selected sources, and the extracted data corresponding to the research questions.

3.1.5 Data Synthesis

After the collection of sources' identification data into the data extraction table. The sources were read or watched, and the corresponding research question data was extracted by reading or watching the sources, analysing, and interpreting the content to extract items for research questions, and finally classifying the found items for conducting overviews. This approach enables categorizing data from various sources as general ideas, and to elaborate on them with specific examples from sources.

To achieve a more accurate mapping and understanding of where the issues occur, the thesis employs data pipeline stages of Extract, Transform and Load to each issue, symbolizing the classic ETL pipeline approach. A fourth category overseeing the entirety of the pipeline is also included for sources that do not exclusively mention the stage or consider the challenge or solution to have an effect in the entire pipeline.

3.2 Conducting the Search

This section focuses on reporting the results according to the search procedure introduced previously.

3.2.1 Search

Ebsco Discovery Service search returned 62 results. Most duplicates were removed immediately by the service itself. After applying the exclusion criteria using the advanced search functionality, 40 results remained.

Google Scholar returned 6320 results. While reading the papers, it quickly became clear that the results got exponentially inaccurate after each page in terms of mentioning data quality within data pipeline. The author restricted the quality assessment to 100 first results. The cutoff point was chosen because of the lower inclusion percentage after 70 results, and since latter papers required constant reading past abstract, introduction, and conclusion to simply

determine their fit for inclusion. After applying the exclusion criteria to the 100 first results, 78 results remained.

The number of results returned by YouTube was not available. To limit grey literature, the author established a cutoff point at 38 results due to the titles and descriptions of the entries extensively drifting from the search string. After applying the exclusion criteria to the 38 most relevant results, 30 results remained.

Google returned 160 000 results. The author established a cutoff point at 30 results where the titles and descriptions of the entries started to increasingly drift from the topic of the thesis. After applying the exclusion criteria to the 30 most relevant results, 16 results remained.

A more detailed number of results after applying each of the exclusion criteria is presented in Figure 3. After exclusion, the sources were read or watched to assess their inclusion and accordance with the quality criteria.

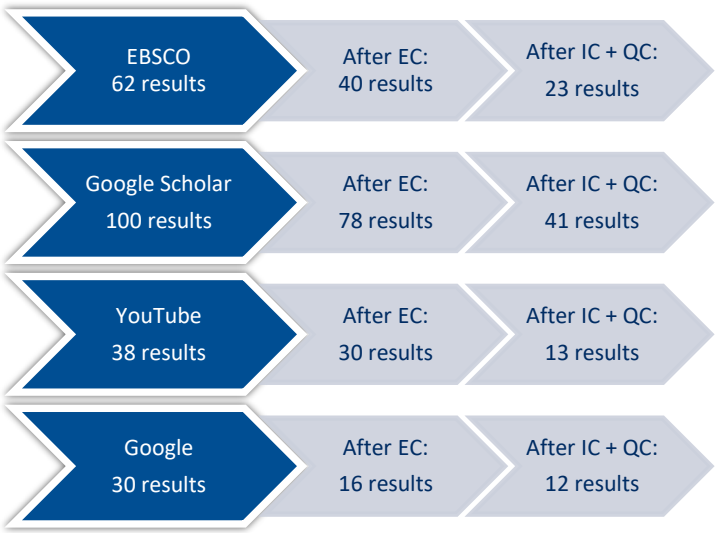

Figure 3 Results After Quality Assessment

After exploring the remaining results, the inclusion criteria and quality criteria were applied. For written sources, the abstract, introduction, methodology and conclusions were read. If the source was not directly focused on data quality within data pipelines, relevant paragraphs containing the search terms were read. In many cases, the full text was read to determine the suitability for inclusion.

For EBSCO, after reading the sources, 23 results remained.

For Google Scholar, after reading the sources, 41 results remained.

For YouTube, after watching the sources, 13 results remained.

For Google, after reading the sources, 12 results remained.

The final number of results selected for data synthesis was 89.

3.3 Demographics

The search and the selected sources provide useful information about data quality challenges in data pipelines in terms of authors involved and yearly distribution. The proportions regarding academic and grey literature are informative as well.

Figure 4 shows that 62 of the selected sources were peer-reviewed academic papers, and 27 were carefully selected complimentary grey literature sources. The volume of grey literature is low enough for it to not compromise the results, and still provide additional insight on the subject.

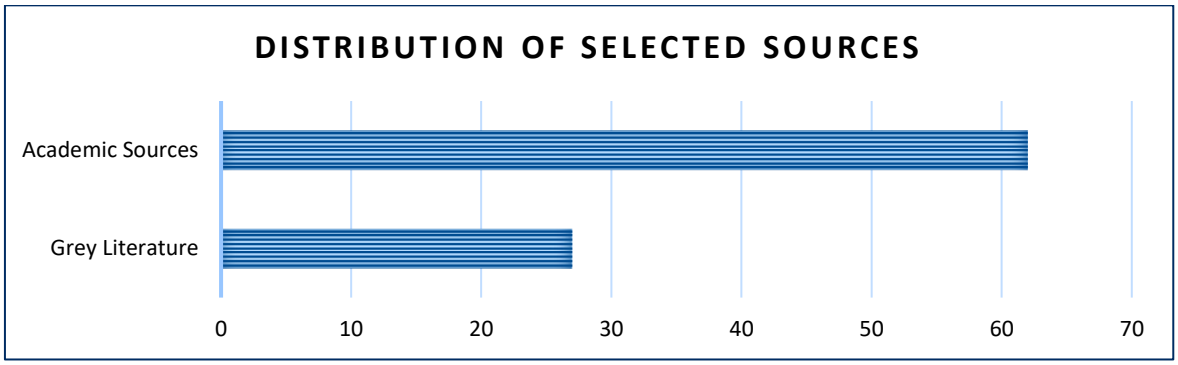

Figure 4 Distribution of Selected Sources

Figure 5 shows authors engaged in more than one paper and symbolize a higher relevance in the subject. Aiswarya Raj Munappy and Jan Bosch led the most relevant academic studies on the matter, and their work was always very closely related to data quality aspects within data pipelines. Álvaro Valencia-Parra focused mostly on big data pipelines and participated in four papers. Wayne Yaddow contributed through grey literature, and his work focused very much on general challenges. The rest of the authors represented in Figure 5 can also be considered closely related to consistently mentioning important data quality aspects associated with data pipelines, and many of them also worked together with Aiswarya Raj Munappy and Jan Bosch.

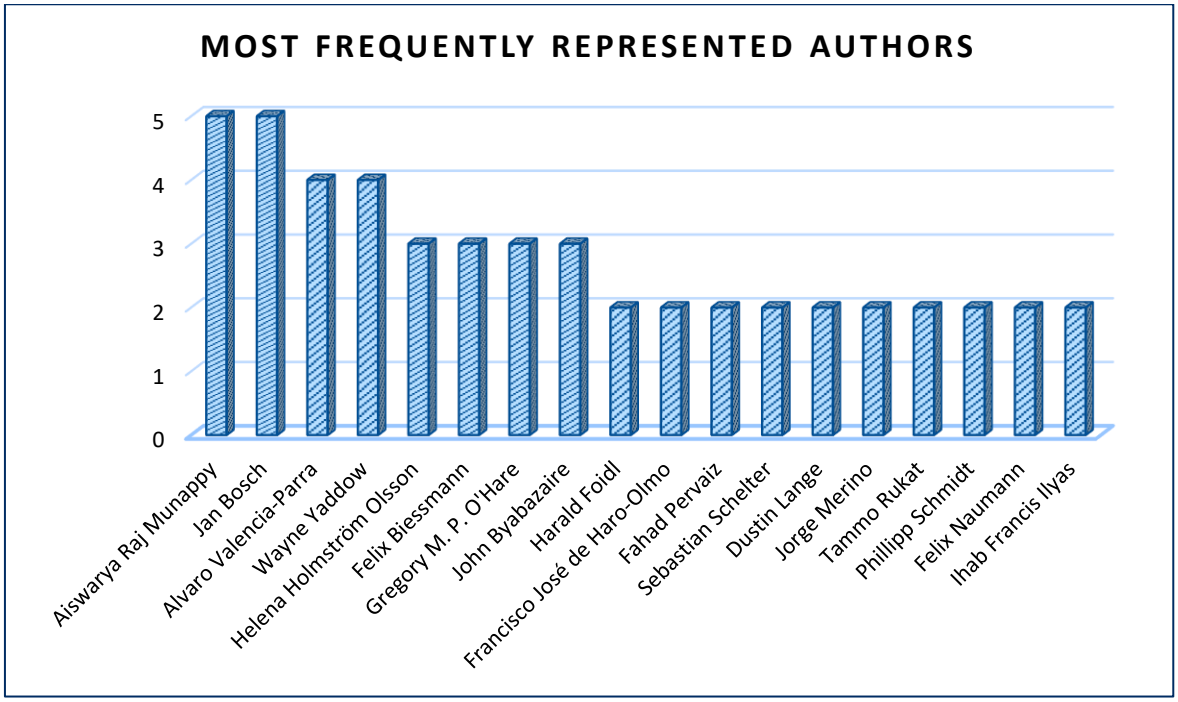

Figure 5 Most Frequently Represented Authors

Figure 6 confirms the good timing of the research, as most of the sources are from the last five years. Data pipelines have been gaining popularity since 2012 yet data quality focus was not initially included as frequently, until 2019.

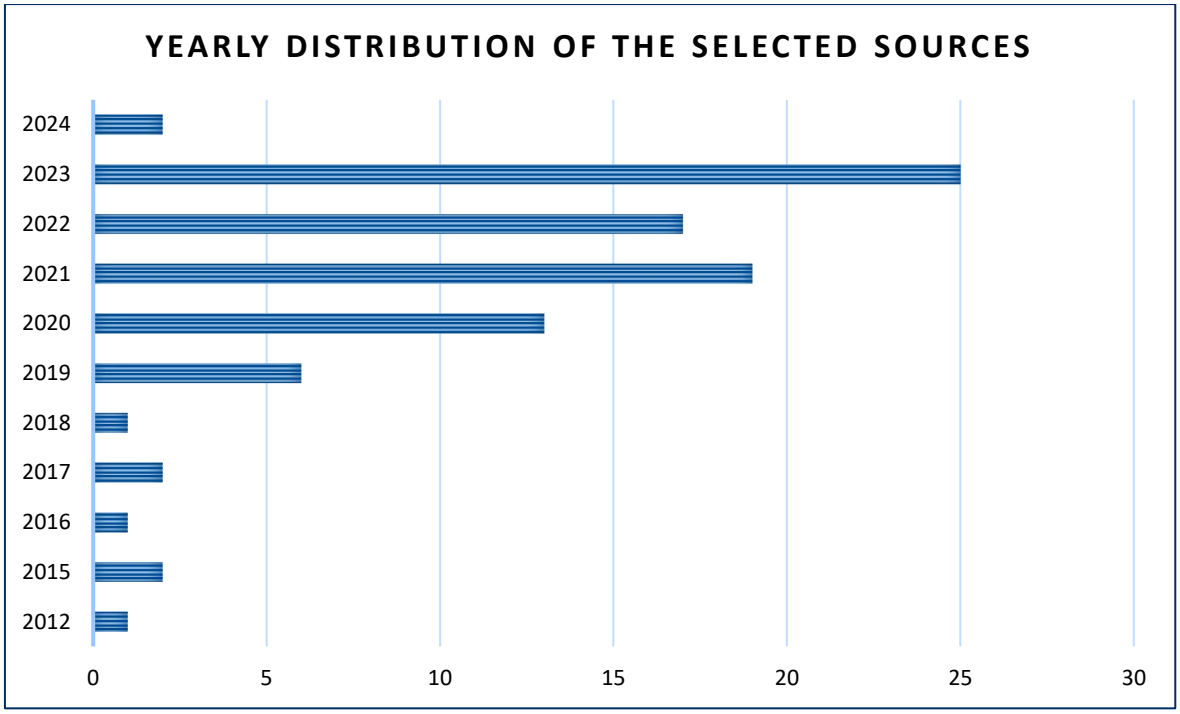

Figure 6 Yearly Distribution of the Selected Sources

4 Results

The results are divided into three separate sections of challenges (RQ1), solutions (RQ2) and open issues (RQ3). There was a total of 219 mentions of issues from 89 sources affecting data quality within data pipelines. Collecting each issue as an individual mention rather than making note of the source allowed a more effective workflow, as the final classification of each issue entity was done after data extraction. Figure 7 shows the overview of the 219 instances of data quality challenges found from the sources and their respective classification.

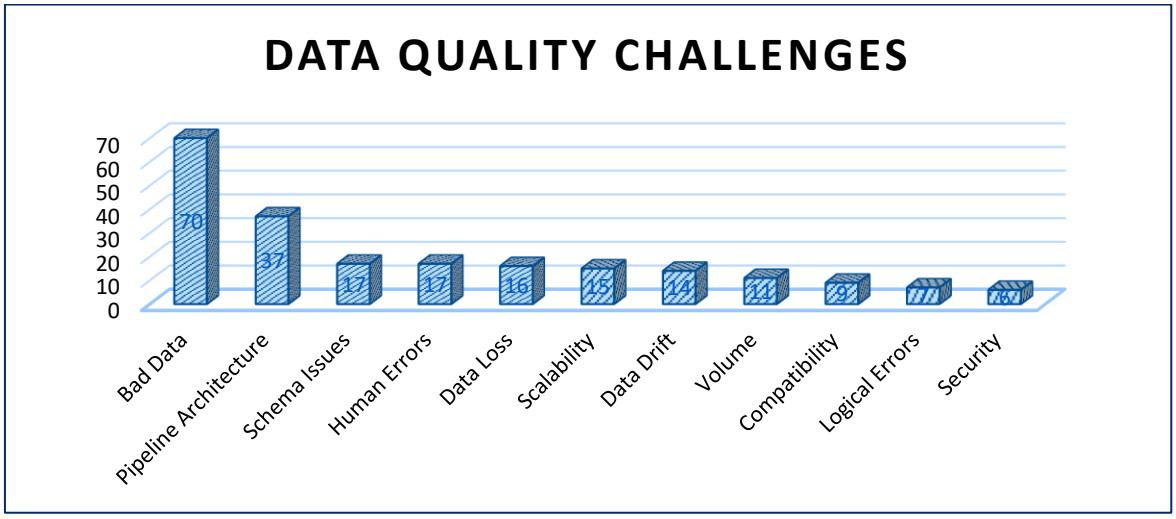

Figure 7 Data Quality Challenges Within Data Pipelines

4.1 Challenges

The found challenges are classified into eleven first-level groups, with each group managing their respective second level subclassifications. Figure 8 shows the two-level classification of the found challenges. Each group has an additional orthogonal ETL stage classification, to allow a deeper mapping and understanding of the challenges.

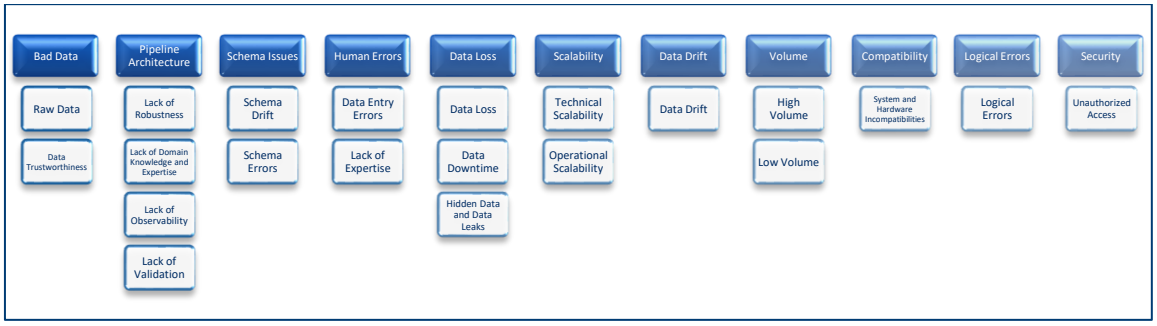

Figure 8 Classification of the Data Quality Challenges

4.1.1 Bad Data

Data quality issues caused by bad data were by far the most popular with 70 mentions (32%). These challenges were difficult to break down into smaller groups, as many sources were not always clear about deep-root causes, or the issues were very general in their nature and consisted of too many elements. The data quality challenges revolved around either the raw nature of the data, or the lack of trustworthiness with the data.

4.1.1.1 Raw Data

There were 35 mentions of raw data issues (16%). The data that enters the pipeline is often diverse due to the heterogeneous nature of data sources. This diverse nature of data causes most problems during data ingestion and pre-processing, but also during the actual transformation phase, where the bad data is already extracted and needs to be handled. Table 2 presents a distribution of sources discussing raw data challenges found for each stage.

Table 2 Distribution of Raw Data Challenges

Extract (22)	**Transform (8)**	**Load (0)**	**Entire Pipeline (5)**
[18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39]	[25], [26], [30], [38], [40], [41], [42], [43]		[19], [42], [44], [45], [46]

a) Extract Stage

Merino et al. [30] consider heterogeneity and incompleteness of data sources to be problematic in data quality evaluation early in the data pipeline. Two different sensors can report hugely misaligned data, suggesting inconsistency and creating linking issues with the data [30].

In analysing data from IoT sensors, Haro-Olmo et al. [20] find that the diversity and heterogeneity of the data sources raise significant challenges with integrating data into big data pipelines. While data pipelines are designed to have the extraction phase to solve data diversity and heterogeneity issues, the problem arises with determining whether the data is even of high enough quality before entering the pipeline. Not only does the data need to be pre-processed before integration into the pipeline but its coherence needs to be evaluated as well [20]. In some cases, the sensor data can be considered low-quality (i.e., incomplete data, wrong readings, null readings) and must often even be discarded as it introduces errors and general obstructions [20].

Lee et al. [21] address the heterogeneity and incompleteness in mobile sensor data for short-term depression detection. The problems regarding data completeness and accuracy of the sensor data were related to user engagement and technical problems [21]. Data imbalance was also introduced on depressive mood within users [21]. These problems are directly affecting data ingestion and can introduce data quality issues in the latter stages of the data pipeline. In [21] data pre-processing removes the clearly erroneous and redundant data points but processing heterogeneous and incomplete raw data can still introduce data quality issues.

Haro-Olmo et al. [24] discuss the challenge with large amount of heterogeneous data during data integration in the pipeline. There is the need to effectively aggregate the heterogenous data (data curation), while ensuring data quality, and to evaluate the quality of the data within data curation [24].

Valencia Parra [28] mentions the issue of integrating heterogeneous data from various data sources with different semantics and schemas, and this raw data issue becomes even more relevant as the volume, variety, and velocity of the data entering the pipeline increases.

Businesses can collect data from different applications, platforms, and databases [19]. As the data sources increase, collecting the same data from separate sources can result in duplicate data, which can in turn skew analytics and metrics [19]. The duplicate value problem is also found by [32], which finds that syntactic issues of duplicate entries or invalid values to be difficult to detect.

Ilyas and Naumann [26] mention raw data sources as a data pipeline challenge. The methods in which the data is obtained, namely grading tasks of humans, sensor readings and extractions scripts, are the root entry point of low-quality data into the data pipeline [26].

Ardagna et al. [33] mention data integration issue with record linkage in big data pipelines. This is an issue where data sources have defined different identifiers for the same data item [33].

Roman et al. [23] identify differently calibrated and different models of sensors to produce misaligned solar panel sensor data. Major obstructive changes in the experiment environment in data sources can also produce data that produces an issue with data quality into the pipeline [23].

Taneja [34] mentions incorrect delimiters and randomness of the columns in raw data to cause issues with loading the data into the pipeline.

Merino et al. [30] mention differences in sensor data completeness to be an issue within data sources, and during data collection and preparation.

Mattila [38] discusses data smells, caused by poor practices of data management and engineering. Data smells can be introduced within data sources during data generation but also during acquisition and processing [38].

Biessmann et al. [37] highlight issues with data heterogeneity and lack of standardization during data collection. IoT sensors often need standardized semantics regarding the equipment, and data quality measures to solve uncertainty [37].

Valencia Parra [29] highlights the heterogeneity of data sources as a separate challenge caused by lack of standardized models in big data context. A similar issue is highlighted by [25], with the challenge of dealing with various standard formats when standard tools lack support for all these formats.

Biessmann et al. [37] mention missing values being one of the most frequent data quality issues. Missing values can break data pipelines and cause issues with data completeness [37]. For example, missing values within datasets can cause data pipelines feeding into ML systems to act irregularly [32]. Missing values can also be introduced through sensor malfunctions in raw data [39].

Golendukhina et al. [22] highlight the high heterogeneity in raw data, in addition to data smells. The lack of well-known standards for data collection and preparation methods is not simplifying the matter. Missing value inconsistencies (*NaN* vs *None*) are mentioned as a consistency smell in [22]. Syntactic smells include ambiguous values (abbreviations) and homonyms (Boston is in USA and Australia). Believability smells include incorrect longitudes and latitudes (missing decimals), incorrect distances (calculated from inaccurate coordinates) and incorrect duration values (time zone issues). An example of a data smell introduction into the data is differences in the methods of data management and handling in data sources. [22]

Jäger et al. [31] signify missing values as one of the more frequent data quality problems that can break data pipelines. This is often an effect of heterogeneous data [31]. Similarly,

Munappy et al. [27] highlight gathering data from multiple sources to be challenging due to missing data, inconsistent data flow, and incomplete data.

Aaron [19] mentions incomplete, inconsistent, and ambiguous datasets to pose challenges for data pipelines, as missing or incorrect records in key areas can hugely affect data quality. Conflicting formats is one way of introducing inconsistent data to the pipeline [19].

Data is not always in the required format and problems arise when the source data value is not its intended format [18]. Data extraction methods cannot scrape and process the data correctly if the type of the value does not match its intended type or is incorrectly formatted.

Golendukhina et al. [22] recognize encoding smells as a problem of medium severity. These include DateTime being formatted as string that complicates conversions with time zones and can cause a loss of information in the data [22]. More issues occurred with numbers as strings, as numerical operations were compromised resulting in potential information loss again [22].

Data can exist in various formats, different databases, and cloud services [35]. The heterogeneous nature of data poses a threat to data completeness if data sources cannot be read fully [35].

Aaron [19] mentions the lack of structure in data as an issue during data ingestion and aggregation. If the data does not fall in the pre-defined format and contains different data types, it cannot be used efficiently [19].

Yaddow [36] mentions format inconsistencies of data to cause problems in data aggregation and analysis. This causes difficulties with creating a unified view of data health and quality [36].

b) Transform Stage

Wrembel [40] mentions data sources, by virtue, to be heterogeneous and dirty. The main complexities lie in cleaning and deduplication and since the source data can be erroneous, incomplete, and inconsistent, it is not simple to find the right algorithms for a data deduplication pipeline [40]. Common data quality issues regarding developing a pipeline to handle data from low-resource environments include duplicated data, inconsistent values, merge data, sync data, format conversion, anomaly detection, variable calculation, data fabrication and analysis [42].

Pervaiz et al. [25] mention that 80% of data processing time is spent on cleaning and aligning data. The main concerns regarding data quality are errors, duplicate values, and inconsistencies of merging data from multiple sources [25]. Researchers have laid out solutions regarding data warehousing, however their works target well-structured idealisms more than the real resource-constrained setting of developing world, where datasets are deemed to be less organized [25]. This means that data processing still needs to account for and work with raw and imperfect data, aiming to either increase or maintain its data quality.

From data cleaning perspective, [26] identifies outliers, constraint violations, duplicates, and missing values as potential challenges. Outliers suggest differences from all other elements, suggesting an unintended value. Constraint violations are already implemented quality checks that have failed, resulting in unintended value yet again. Duplicates represent elements of the same definition with different representation possibilities, causing inconsistencies in data. Missing values simply represent lack of data value but in more complex scenarios it creates a lack of understanding behind the intention of the empty or

null value. The fundamental challenge arises when these issues are unresolvable within the data sources, it requires the pipeline to function with the available data. [26]

Merino et al. [30] mention data uniqueness, number of duplicates and constant values as problematic aspects during data pre-processing. Semantic accuracy (i.e. data closeness compared to reality) is another problem that can cause problems with the overall data accuracy [30].

Dazzeo [43] mentions the variety and accuracy of data to be a problem during data transformation. The data can come in different formats, with issues related to human error, measurement limitations or suboptimal data collection [43]. This issue is also related to the heterogeneous nature of the source streams, making it a challenge to perform transformations to make the data homogeneous [43].

Huang et al. [41] explain label scarcity as a labelling problem in the transformation stage of building a ML data pipeline. The main inefficiency problems are caused by complex data in the form of dashboards, log bundles and configuration files [41].

c) Load Stage

There are no sources mentioning challenges related to raw data in the load stage.

d) Entire Pipeline

Munappy et al. [46] acknowledge real-world raw data to often be incomplete, inconsistent, and/or missing certain behaviours. With raw data often also being erroneous, there is instantly a challenge with producing high quality data in the data pipeline. Pervaiz [42] has also recognized handling inaccurate, incomplete, and inconsistent data as a challenge that data pipeline must solve.

Osborn [44] identifies null or blank values to be one of the more common data quality issues. These values are inherited from data sources or introduced in errors within the pipeline [44]. Duplicate data from loose data aggregation or typing errors can also introduce problems from ingestion to storage to influence data quality [44].

Data in different languages can introduce an additional challenge in ensuring data quality [19]. Language challenges may manifest in the ordering of first and last names across various countries and saving that can reverse the names [19]. Translations and differences in units might also introduce inaccuracies in unifying the data [19].

Zou and Xiang [45] describe a big data quality problem with extracting immense volume of diverse data content with complex data structure. There are inevitable data quality issues of errors and missing values in big data and using substandard quality data will lead to inaccurate data pipeline output [45]. Big data quality assessment has been researched but they all contain limitations in evaluation of the data within data pipelines.

e) Summary

The raw nature of the ingested data is a broad problem, but there are some subclasses of issues that can be found in the previous analysis. These issues are difficult to classify, because they are often too interlinked and coherent to be separated from each other. Based on the previously found raw data challenges, the following are the most frequently mentioned factors that contribute to challenges within data pipelines:

- heterogeneous and diverse nature of data,
- incomplete, inaccurate, and inconsistent data,

- lack of standardized models and formats,
- errors and missing values,
- duplicate data and inconsistencies,
- syntactic issues and data smells.

4.1.1.2 Data Untrustworthiness

There were 35 issues (16%) related to lack of trust within the data. The factor of data trustworthiness in data pipelines plays a significant role. Not all data is as valuable or as accurate, and data pipelines often struggle with assessing the trustworthiness of the data. The absence of quality metrics in the data further complicates quality-driven decision-making throughout the pipeline. This issue is most present during data intake, but also throughout the entire pipeline and its transformational phases, where the pipeline desperately needs the data to have some form of measurement for its trustworthiness. Table 3 presents a distribution of sources discussing data untrustworthiness challenges found for each stage.

Table 3 Distribution of Data Trustworthiness Challenges

Extract (22)	**Transform (6)**	**Load (2)**	**Entire Pipeline (5)**
[13], [19], [20], [27], [29], [42], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [39]	[35], [62], [63], [64], [65], [66]	[33], [67]	[30], [36], [44], [68], [69]

a) Extract Stage

Addressing the general challenge of extracting valuable information from extensive raw data is crucial, as emphasized in [47]. Effective decision-making needs support from both high-quality data information and its thorough assessment, underscoring the critical role of data reliability [47]. Rahman et al. [48] contribute valuable insight to the issue, specifically addressing the lack of confidence in the reliability of traffic detector data and the absence of tools for automated quality testing.

For [20], there is agricultural humidity and temperature data collected from various type sensors for both offline and online scenarios. Without a meaningful measure of quality, decreased data quality could be introduced as early as the data is collected. If low-quality data reaches the transformation phase of the data pipeline and it is left unevaluated or without a proper label, then it introduces an instant problem with further data quality. As stated by Haro-Olmo et al. [20], the data needs to be evaluated before integration.

Government agencies collect diverse data to comprehend development indicators, whereas smaller organizations can collect data with more specific insights [42]. This results in disorganization, inconsistencies, isolation and lack of structure and standards when all the data is extracted to a silo [42]. More reasons for poor-quality source data can be the reason of lack of qualifications and training of people and entering data from paper forms. Pervaiz [42] focuses particularly on a challenge with producing high-quality data from development data collected in low-resource environments, emphasizing literacy, infrastructure, culture, partnerships, and funding to be the barriers. Recording and digitizing, and unsupported requirements are mentioned as concrete challenges [42].

Typically, data quality is evaluated prior to data utilization. This can be challenging in various machine learning scenarios. ML algorithms consist of three intricately linked components: model representation, accuracy evaluation measures, and model optimization methods. Because of this interconnectedness, evaluating data quality for ML applications is not straight-forward and it poses a challenge in creating appropriate quality assessment processes within data pipelines. [27]

The fundamental elements of data acquisition, including signal transmitters, carriers, samplers, converters, and networks, significantly influence both the data collection process and the quality of the gathered data [49]. Inconsistencies in data may arise from various sources, such as electromagnetic interference, ground loop errors, power surges, jitter, signal aliasing, quantization errors, and analog filtering [49].

Abdellaoui et al. [50] mention sources using traditional DISO formalization which does not follow data quality concepts, which contributes towards the challenge of evaluating data quality properly.

Foidl and Felderer [51] mention data source quality issues with data models, schemas, and values. The problems include spelling errors, domain or business rule violations, lack of validation, and missing or duplicated values [51]. This intensifies data quality challenges within data pipelines as low-quality data needs to be properly handled and validated away from its source.

Aaron [19] identifies lack of validation as a problem that can enable bad data to enter the pipeline. Redyuk et al. [70] establish the same problem with large batches of data containing erroneous data, which corrupts the validity of the whole batch. The bad data in [70] is simulated with explicit and implicit missing values, numeric anomalies, swapped numeric and textual fields, and typos.

Deshpande [52] discusses the "good pipeline, bad data" problem, which stems from lack of data *trustability* allowing bad data to flow through the pipeline. Data is hard to trust when it has no metadata or lacks origin information [52]. Third-party data sources are often prime examples of introducing untrusted bad data into the pipeline [52].

Valencia Parra [29] raises issues with uncertainty and accuracy of data during data integration in big data environments. Automatic data integration can make assumptions and although probabilistic models allow assessment and countermeasures, this does not scale well for big data systems [29]. The main challenges with improving accuracy include conducting its proper assessment and developing optimization techniques [29]. The key to this is data provenance, which comes with additional challenges related to versioning, size of the metadata, interoperability, metadata queries and reproducibility [29].

Taverna et al. [53] point out that data can have incomplete and inaccurate metadata. It is challenging to include the same metadata with each data entry, and this causes challenges when estimating data quality metrics within the pipeline [53]. Similarly, [54] finds missing metadata information to be one of the unsolved challenges in medical data pipelines. Matching entities, inheriting metadata, and retaining data integrity are the related data quality challenges with medical data [54].

Hu [55] highlights a problem with teams not having the proper metadata they need to accurately assess the quality of data. The pipeline can have extensive dashboards and reports to assess data quality, but it is difficult to produce assertive results with incomplete data [55].

Data can be corrupt or inaccurate, and [56] marks this as a veracity problem within ML pipelines. Gathered data can be inconsistent and have no measure for certainty, making it difficult for data engineers to collaborate with it [56].

Smart farming uses machinery, equipment, and most importantly - sensors, which have variable precision, ambiguities, and poor interoperability [57]. This is a data quality issue during data ingestion, as the data is not consistent and does not have a quality measure.

Kauhanen [58] mentions issues with data observability within Finland's railway operation domain. There is no visibility of all the diverse types of data entering the pipeline, and it causes data quality issues [58].

Data sources and providers can often introduce *trustability* and data quality concerns, and the verification process often exceeds human capacities [59]. Data quality characteristics include accuracy, availability, and consistency, and they often require in-depth analysis to evaluate [59].

Sacolick [60] talks exclusively about data debt, and its introduction via low usage, inferior quality, or underwhelming data into data pipelines. There is also a growing issue of bad data quality in marketing, with data quality rendering marketing measurements useless [61].

Munappy et al. [13] offer extensive guidelines for modelling data pipelines. However, data quality challenges cannot be completely solved because if the data produced in the source is low quality, there is no good mechanism to suddenly improve the quality [13].

Zhang et al. [39] highlight that given the collection of sensor data in a free-living environment, inherent noise in wearable sensor signals becomes inevitable. The quality and coverage of sensor data are linked to dBM derivation, thus playing a crucial role in shaping overall data quality [39]. Furthermore, generating measures to assess the data quality in the sensor data is not straight-forward either and can be quite error-prone [39].

b) Transform Stage

With a lot of freely accessible data and the growing amount of semi-structured and unstructured data, the quality and trustworthiness of the data is deemed to be in question. Since quality assessment of big data is integral to architecture design, shaping the data processing stage in a data pipeline within a big data system can be challenging. Data from suspicious sources, or corrupted, subjective, inaccurate, or incomplete data can go on to influence the data processing. Challenges also arise as data quality relies on contextual factors, making evaluation complicated as the same quality attribute may vary in different situations, requiring distinct evaluation metrics. [62]

Sadiq et al. [65] mention untrustworthy data sources to cause problems with data transformation within data pipelines. In-field measurements in photovoltaic data systems often produce invalid data in the form of gaps, missing values, and erroneous values [64]. This is caused by power outages, equipment faults, communication issues or maintenance interruptions [64]. Invalid data cannot be processed effectively [64].

Wallace [35] writes of inaccurate, incomplete, or inconsistent data within data quality assessment to be one of the key issues within pipelines. The challenge lies within developing solid data validation, cleansing and quality assurance mechanisms [35]. The transformation requires intricate transformation techniques and frameworks to work with raw data, and to ensure data quality [35].

Klievink et al. [63] describe an issue with increasing global trade where cargo data availability information is not timely and accurate and lacks visibility. Whereas, a concept-

level data pipeline is proposed to solve these challenges, they still need to be addressed in development [63].

May and Fuller [66] identify dirty data as one of the costliest issues within businesses. In data pipelines, dirty data is problematic to clean, and usually needs significant domain knowledge for data scientists and business intelligence engineers to manage [66]. The worst case is dirty data providing incorrect reports, predictions, and results from the data pipeline, compromising its whole purpose, and leading to financial losses [66].

c) Load Stage

Pulkka [67] raises a prominent issue with *trustability*. There is a lack of data quality measurements on outgoing data and consumers will need insurances that they can trust the correctness of the data [67].

When a receiving system has high expectancy towards data quality, it is important for users to understand the produced data [33]. The interconnections between data flow, surfaced issues and analytics are important and represent the usability of data [33].

d) Entire Pipeline

Scott [68] signifies the looseness and improper implementation of processes to cause the creation of bad data in both data sources and the pipeline itself. Date values inserted in excel without proper field validation can manipulate dates, and inserting these dates into database as strings can change its data type entirely, resulting in data of poor quality entering the pipeline [68]. Logical steps within all stages of pipelines can cause data of inadequate quality to emerge. As bad data moves forward without resolution, it creates more problems, and the core issue becomes harder to isolate [68].

Yaddow [36] mentions handling bad data with advanced data quality checks to often require deep understanding of the data and its management. Identifying false positives and negatives in domain-specific data can be difficult to achieve [36].

Yaddow [69] mentions null values and duplicated values to be a cause of data errors within all data pipeline stages. Osborn [44] mentions the possibility of referential integrity being compromised within data pipeline processes or incoming data.

Merino et al. [30] mention outliers and noise as a data quality concern in all stages of big data pipelines, affecting precision, accuracy, consistency, completeness, and timeliness.

e) Summary

Data not having metrics to signify their reliability and trustworthiness is a worrying sign for data pipelines. Although this is the core issue, there are many factors contributing towards this, and they are interconnected in their nature. The following are the recurring elements that contribute to data trustworthiness challenges within data pipelines:

- uncertainty and missing quality metrics,
- lack of validation,
- incomplete data and missing metadata,
- dirty data, outliers, and noise,
- poor data quality.

4.1.2 Pipeline Architecture

Data quality issues occurring due to pipeline processes, functions, operations, and design decisions were mentioned 37 times (17%). These are issues that are more related to the pipeline architecture, rather than the data itself.

4.1.2.1 Lack of Robustness

Data pipeline robustness issues were mentioned 10 times (5%). When the pipeline has a chance to break down on any scale, there will usually be an impact on the data quality, either through data loss or data corruption. Table 4 presents a distribution of sources discussing challenges related to lack of robustness for each stage.

Table 4 Distribution of Robustness Challenges

Extract (0)	**Transform (3)**	**Load (1)**	**Entire Pipeline (6)**
	[36], [50], [71]	[72]	[13], [46], [53], [73], [74], [75]

a) Extract Stage

There are no sources mentioning challenges related to lack of robustness in the extract stage.

b) Transform Stage

Changes to data processing logic or monitoring can introduce data errors or inconsistencies, leading to unreliability and data quality issues [36].

Husom et al. [71] mention robustness and achieving high data quality as challenging factors in building their ML pipeline with unsupervised learning. The pipeline should withstand raw data processing and training, while having robust data cleaning processes and data versioning [71]. Another complication in designing a robust pipeline was the technical debt within the data pipeline architecture [71].

Detecting and rectifying inconsistencies primarily relies on low-level rules and programs, demanding substantial user involvement [50]. Data pipeline functions are tasked with detecting these violations, yet it is not always possible [50].

c) Load Stage

Unidentified failures during transformation to leave data corrupt, unreliable, and incomplete when it is loaded into a data lake [72].

d) Entire Pipeline

Munappy et al. [46] mention a general situation where a bug in a single step of a data pipeline has cascading effects on data quality. Even the smallest changes in data sources and pipeline architecture or functions can initiate negative effects in the continued path of the data [46]. Similarly, there are issues with degradation of data quality as data moves through the pipeline [46].

Munappy et al. [13] mention a problem with reliability. When operations in the pipeline are not fault-tolerant, and are built without validation and mitigation mechanisms, they cannot ensure data quality [13]. Munappy et al. [46] point out a common challenge related to the intricacy and the challenge of identifying all potential faults across various stages of the data

pipeline. Designing a completely fault tolerant data pipeline is difficult from software developers' perspective.

Bail [73] emphasized testing of data to ensure data quality within the pipeline. This is a difficult task in terms of knowing which parts of data to test, implementing countermeasures, keeping these components up to date and providing some sort of quality measure for the data [73].

Somasundaram [75] points out the complexity of monitoring infrastructure to cause coordination and management issues with real-time monitoring. Maintaining fault tolerance, providing visualization, and developing actionable insights is all part of keeping the monitoring running effectively and not causing data quality issues [75].

Improper message reconciliation is a key problem in data pipelines, causing inconsistencies and timeliness issues with data quality [74].

Taverna et al. [53] point out the evolving nature of standards and format versions. This can cause data pipelines to fail when standards and formats are updated [53].

4.1.2.2 Lack of Domain Knowledge and Expertise

The lack of domain knowledge and expertise while developing data pipelines was mentioned 10 times (5%). Data quality is also heavily influenced by its domain and systems between which the pipeline is operating. When the domain or its needs are too complicated, data pipeline will struggle to maintain data quality. Table 5 presents a distribution of sources discussing challenges related to lack of domain knowledge and expertise for each stage.

Table 5 Distribution of Domain Knowledge and Expertise Challenges

Extract (0)	**Transform (3)**	**Load (1)**	**Entire Pipeline (6)**
	[25], [55], [76]	[33]	[29], [52], [54], [55], [66], [70]

a) Extract Stage

There are no sources mentioning challenges related to lack of domain knowledge and expertise in the extract stage.

b) Transform Stage

Data engineers not being able to understand domains and data specifically enough creates a problem with establishing high data quality within pipelines [55]. This problem is further amplified by the lack of experienced data engineers [55].

Clayson et al. [76] underscore the significance of methodological choices in impacting data quality within the data pipeline. It emphasizes the use of internal consistency estimates as a proxy for data quality and the need for a more direct examination of metrics like between-trial standard deviations [76]. The processing stages, including condition selection, filter cutoffs, and electrode site usage, summarize a complex situation where data quality challenges may arise, urging a need for an optimized data processing pipeline [76].

Pervaiz et al. [25] mention an important problem with many tools saturating the data processing scene. This complicates inner operations within a pipeline and requires more merging and analysis, inducing problems at later stages of processing pipeline [25]. Data

passing through multiple tools can have implications for its quality, making it challenging to process at certain processing stages.

c) Load Stage

The accuracy of the data is heavily dependent on the specific needs and requirements of the users [33]. Choosing the right analytics and infrastructure based on requirements and previous deployments plays a huge role in increasing the accuracy of computations [33].

d) Entire Pipeline

The domain and semantics of the data heavily influence the quality assessment module within a data pipeline [29]. Maintaining data quality across the data pipeline can pose challenges, particularly concerning the five quality dimensions outlined in big data [29, 77]. The main issue lies in devising metrics for each dimension, along with automatically suggested actions [29].

May and Fuller [66] signify the problem with outsourcing data pipeline development. This issue stems from lack of domain knowledge and will oftentimes result in non-optimal ways of ensuring data quality [66]. Additionally, [52] covers the problem companies and their pipelines lacking roles to cover issues related to data maintenance and data quality assurance. Data engineering is blowing up, yet data is still seen as a secondary entity in software development and this, among other problems, causes issues with data observability within data pipelines [52].

Redyuk et al. [70] mention incomplete domain knowledge to cause bad architectural decisions when building the pipeline. Even domain experts might not be able to comprehend the full scope of all the pipeline processes, which causes the pipeline to perform poorly due to false alarms and missed errors [70].

Rahman et al. [54] signify the importance of domain expertise in designing data pipelines. Domain experts are expensive resources, and their overwhelming usage might cause data pipelines to be built in an overly customized and non-scalable way.

The wide range of engineering tools and downstream application use cases amplifies the spectrum of challenges faced by data engineers to difficult levels [55]. Performing thorough data quality assessments becomes challenging due to the inherent risks introduced by each tool, even though they may initially address the primary issue [55].

4.1.2.3 Lack of Observability

Data lacking observability while moving within the data pipeline was mentioned to cause data quality issues 10 times (5%). Many data quality issues stem from the fact that the pipeline or the data is not observable. This allows faulty processes to operate without repercussions, and bad data to pass through without any way of interacting with its quality. Data intricacies, as well as all the pipeline operations, need to be traceable and observable. Table 6 presents a distribution of sources discussing challenges related to lack of observability for each stage.

Table 6 Distribution of Observability Challenges

Extract (0)	**Transform (3)**	**Load (0)**	**Entire Pipeline (7)**
	[49], [69], [75]		[36], [49], [52], [78], [79], [80], [81]

a) Extract Stage

There are no sources mentioning challenges related to lack of observability in the extract stage.

b) Transform Stage

During pre-processing, one common challenge is dealing with gaps in the data [49]. These gaps can occur for various reasons, such as sensor malfunction, communication errors, or simply due to the intermittent nature of data collection and the length of these gaps can vary significantly, depending on the dynamics of the monitored phenomenon and other contextual factors [49]. Addressing these gaps properly is crucial for ensuring the integrity and reliability of the dataset before further analysis. Data quality issues can also arise during transformations, cleansing, and enrichment [69].

A data versioning problem within the data pipeline arises when data is changed by an algorithm during pre-processing, and it is later difficult to know which data was changed and what algorithm was used [49]. Lacking a comprehensive data history makes it challenging to uphold data quality when encountering issues that require attention during later stages of processing.

Somasundaram [75] found complicated data transformations of filtering, aggregations and join operations to require constant monitoring and validation to ensure data integrity.

c) Load Stage

There are no sources mentioning challenges related to lack of observability in the load stage.

d) Entire Pipeline

Byabazaire et al. [78] question the subjective assessment of data quality in big data models. Additionally, the forms in which data quality is measured differ depending on the stage and context of the model [78]. There is a challenge in choosing the stages where data quality should be evaluated and fusing quality scores from independent stages to an advertisable data quality score for applications [78].

Moses [52] mentions an issue with data lacking observability, causing teams to spend unnecessary amounts of time fixing the problems.

Ovens [79] signifies the importance the recognizing the overall flow of the pipeline functions. When the pipeline lacks observability, the data can become stale with no possibility for instant reconciliation [79].

Increased complexity within the pipeline infrastructure can cause challenges with observability and pinpointing data errors [36].

Valarezo et al. [80] establish a challenge with producing accurate, reliable, and timely data in our data-driven world. Data pipelines need an approach to define and prove data quality through visualization or metrics [80].

Data historians typically compress data, resulting in potential information loss [49]. When there is need to access the original data, the reconstruction can lead to loss of important statistical features [49].

Ryza [81] mentions data pipelines to sometimes produce incomplete, inconsistent, or erroneous data. Producing and delivering bad data on time is often considered worse than delivering no data at all [81]. Data quality is a difficult concept and with it being a

fundamental issue within a data pipeline, it cannot just be guaranteed by using a data quality tool [81].

4.1.2.4 Lack of Validation

Lack of validation within data pipelines was mentioned 7 times (3%). When data pipelines are built in an overly robust manner, it allows for bad data to flow through the pipeline without any issues. The data flowing through the pipeline needs to be verified through quality estimation, and validation mechanisms and restrictions need to be defined. Table 7 presents a distribution of sources discussing challenges related to lack of validation for each stage.

Table 7 Distribution of Validation Challenges

Extract (0)	**Transform (5)**	**Load (0)**	**Entire Pipeline (2)**
	[13], [22], [24], [49], [82]		[51], [75]

a) Extract Stage

There are no sources mentioning challenges related to lack of validation in the extract stage.

b) Transform Stage

Yao et al. [82] raise an issue with deep neural network fault severity estimation based of mechatronic systems. The challenge is to maintain high data quality consistent with the fault information when building the pipeline [82].

Although data pipelines inspire productivity and increase the quality of data, Golendukhina et al. [22] mention that poorly developed data pipelines will not recognize data quality issues and can even produce data of poor quality. The tools used in data transformation do not sometimes recognize values as time variables or handle them as strings, which is a data smell [22]. Data enrichment is considered as the phase where most smelly data is introduced [22]. There is often no verification that the transformation phase ensures high data quality or if the incoming bad data or data smells are even identified [22].

When algorithms are used with low-quality data, the result will also inevitably be of low-quality [13]. When collecting fault logs, functions cannot distinguish between fault types if some unnecessary parts of the data are removed during pre-processing [13]. When data loses some of its value and becomes insufficient during the transformation process, there is sometimes need for the original raw file [13].

Therrien et al. [49] mention manual fault detection as a time-consuming process during data pre-processing. However, without adequate fault detection, the quality of the data will naturally decrease.

Processing poor-quality data produces incorrect decision-making and damages the performance of functional operations and finding the satisfactory level of usability of data is therefore crucial [24].

c) Load Stage

There are no sources mentioning challenges related to lack of validation in the load stage.

d) Entire Pipeline

Foidl and Felderer [51] discuss the challenging nature of validating data quality, especially within machine learning scenarios due to high complexity of both the data and the systems. Practitioners need to use data quality checks to contain the data quality issue, yet a completely exhaustive validation is not manageable [51].

Somasundaram [75] lists anomaly detection, missing values, and outliers as keywords for data validation in data pipelines, all while retaining data integrity and accuracy within the domain.

4.1.3 Schema Issues

Data quality issues regarding schema drift and schema errors within data pipelines were mentioned 17 times (8%).

4.1.3.1 Schema Drift

Schema drift was mentioned 13 times (6%). Schema drift is an issue where the structure or schema of the source data changes over time. This can happen due to column deletion, data type changes, format changes, or simply schema evolution. This problem was found to be most present in the extraction stage, where schema changes in data sources directly impacted the data entering the data pipeline. On one occasion, schema changes were causing problems during the loading stage. Table 8 presents a distribution of sources discussing schema drift challenges for each stage.

Table 8 Distribution of Schema Drift Challenges

Extract (10)	**Transform (0)**	**Load (1)**	**Entire Pipeline (2)**
[28], [36], [44], [51], [69], [70], [81], [83], [84], [85]		[72]	[52], [68]

a) Extract Stage

With new features and software updates, schema changes are very ordinary [44]. Dropped columns and changes in structure will introduce new complexities and errors to the pipeline, which can in turn result in data quality issues [44]. Schema modifications can result in data mismatches, causing issues with tracking data lineage through monitoring [36].

Song and He [84] recognize a common issue with data columns being altered in upstream data, creating schema drift in the data. These issues are difficult to detect and require considerable amounts of human involvement [84]. The drifts can also happen due to changing schema semantics [28].

Changes within data sources can cause errors with data ingestion and preprocessing [70]. One example of this is changing time measurement from seconds to milliseconds [70].

Armbrust and Lappas [85] highlight changes in data structure and schema errors to be a threat to data quality during data ingestion. Schema changes and the overall schema design are one of the causes for data errors in data pipelines [51, 69]. This is referred to as schema drift, which causes errors in data pipelines [81, 83].

b) Transform Stage

There are no sources mentioning challenges related to schema drift in the transform stage.

c) Load Stage

Vijay [72] has identified schema changes to cause problems with loading the data from the pipelines to data lakes.

d) Entire Pipeline

Scott [68] highlights schema drift in the form of evolving user interfaces or data structures in source data. With pipelines not being aware of schema changes, it becomes an issue with data quality as the pipeline itself is not updated about new ranges or values [68]. This can have a snowballing effect within the pipeline when the pipeline assigns null to the unexpected data item, causing errors in joining operations in the form of null lookups, and providing entirely wrong results to the end user [68].

Deshpande [52] mentions schema drift to affect data quality within data pipelines negatively, with a possibility of even breaking the pipeline.

4.1.3.2 Schema Errors

Schema errors were mentioned 4 times (2%). In addition to schema drift, some sources recognized schema issues in a more general way, where the pipeline is required to process data that violates its designated schema, or where the schema is entirely absent. Table 9 presents a distribution of sources discussing challenges related to schema errors for each stage.

Table 9 Distribution of Schema Error Challenges

Extract (0)	**Transform (2)**	**Load (0)**	**Entire Pipeline (2)**
	[18], [42]		[32], [86]

a) Extract Stage

There are no sources mentioning challenges related to schema errors in the extract stage.

b) Transform

Munappy et al. [18] mention schema errors as one of the common faults during data processing. Additionally, [42] has identified challenges encountered when attempting to reverse engineer schemas within data pipelines.

c) Load Stage

There are no sources mentioning challenges related to schema errors in the load stage.

d) Entire Pipeline

Biessmann [32] recognizes schema violations (e.g., string values in numeric columns) to break ML processing pipelines. In most cases data type constraints raise errors, yet these validations are often counteracted in data pipelines by design as any data type can be cast to string or modified for convenience [32]. For example, a study mentions casting errors induced by excel in biomedical data resulting in one fifth of the papers in leading genomics journals [32, 87].

Raval [86] mentions a general problem in designing data pipelines in which each independent step can introduce defects into data when there are intended changes in schemas and approaches to data handling. The challenge lies within building pipelines in a schema-less way to accommodate high-velocity and variable-length events [86].

4.1.4 Human Errors

Human errors due to human involvement were mentioned 17 times (8%). These data quality issues are related to erroneous human behaviour towards data creation or handling.

4.1.4.1 Data Entry Errors

Human errors caused by data creation were mentioned 9 times (4%). The creation of poor-quality data is mentioned to occur in data sources, leading to subsequent extraction of bad data. Although being a form of bad data, its root cause allows it to seek unique solutions. The data requires either continuous corrective measures, or alternatively, validation protocols set up within the data sources whenever possible. Data entry errors were only mentioned to happen near the extract stage of the pipeline. Table 10 presents a distribution of sources discussing data entry challenges for each stage.

Table 10 Distribution of Data Entry Challenges

Extract (9)	**Transform (0)**	**Load (0)**	**Entire Pipeline (0)**
[19], [22], [25], [42], [44], [49], [53], [55], [68]			

Golendukhina et al. [22] point out a data smell of values entered inconsistently in the data sources because of different people. Therefore, the pipelines' data quality is compromised as early as the data extraction stage. Taverna et al. [53] also mention the chance of human error in composing data, as it can be difficult to produce the same metadata for each data entry.

In under-resourced regions with inadequate infrastructure, large-scale data collection relies on paper forms [25]. For instance, attendance records at rural health facilities in Pakistan are maintained in large paper logbooks [25]. Field workers manually complete these forms, which are then sent to a local data collection centre [25]. Subsequently, data entry workers transcribe the information into a computer, often resulting in poor accuracy [25]. Pervaiz elaborates on this in [42] by mentioning data entry to lack constraint checks, resulting in spelling errors, inconsistent values, and formatting issues. The situation presents a stark contrast when compared to data entry in a system where errors and inconsistencies are mitigated by simple validation [42]. Designing a user interface becomes another challenge in developing tools to solve the data entry problem [42].

Therrien et al. [49] mention manual data collection in offline laboratory analysis to have drawbacks in the form of delays and infrequencies through extensive sample analysis and manipulation, and the process being prone to errors caused by simple human distraction and fatigue.

When data is created by processes that involve humans, there is a strong possibility of data being created by someone who is not a qualified data creator [55]. Data quality is defined before it reaches the data team, and accuracies in the form of abbreviations and typos can

therefore pose downstream data quality risks [55]. This issue is accurately termed as lack of data literacy [55].

Aaron [19] includes human errors as one of the causes for inaccurate data entering the pipeline. These mistakes can happen in the form of typos, format differences, and incomplete or incorrect data entry [19]. Additionally, [44] mentions human typing errors to introduce duplicate data to datasets.

Scott [68] signifies the source of bad data usually being human involvement in the form of data entry errors or omissions. When users have too much freedom in loosely implemented processes, it leads to data quality issues [68].

4.1.4.2 Lack of Expertise

A general lack of expertise with data handling was mentioned 8 times (4%). Although lack of expertise can still produce erroneous data like data entry, the essence of the problem is more related to data quality assessment and decision-making. The data can already be created, and moving within the pipeline, but the lack of expertise in handling it properly can be the source of data quality issues. These issues are related more to where human involvement within the pipeline is happening, rather than to a specific stage. Table 11 presents a distribution of sources discussing challenges related to lack of expertise for each stage.

Table 11 Distribution of Expertise Challenges

Extract (1)	**Transform (2)**	**Load (0)**	**Entire Pipeline (5)**
[62]	[18], [62]		[5], [32], [49], [85], [88]

a) Extract Stage

Immonen et al. [62] recognize the user's changed expectations and visions of data quality as a new challenge in data quality research. Human experts manage qualitative evaluation, and it is based on the visualization of metadata [62]. From this, it can be derived that the bias of humans can influence the assessment of data quality in data sources, therefore making it a potential threat in terms of data quality when it enters a data pipeline.

b) Transform Stage

Human errors are common and data pipelines are designed to minimize human errors in addition to automation [18]. There are some activities within data pipelines that cannot be automated (e.g., data labelling in a machine learning pipeline) [18]. Therefore, human involvement remains and so does the potential erroneous data from it. Munappy et al. [18] also suggest implementing alarms and notifications as a mitigation strategy for problems regarding data loss, however it still considers human involvement to be one of the causes for data errors. It is paradoxical and although it does inflict less harm than good, it is considered an unideal philosophy and excessive human involvement should be avoided. On the other hand, Abdellaoui et al. [62] mention existing automatic repairing solutions to introduce new errors by generating unverified fixes.

c) Load Stage

There are no sources mentioning challenges related to lack of expertise in the load stage.

d) Entire Pipeline

Operational errors are serious issues in non-automated data pipelines and often the process cannot even be fully automated [5]. As they require human intervention, it automatically introduces a chance for human error (e.g., data labelling) [5].

When designing a data pipeline, data scientists often lack the domain-specific mindset to address some data quality aspects [49]. Domain specialists, however, often lack the opposing skillset of what needs to be considered for producing quality data [49]. This conundrum enables design errors to occur when the pipeline is complete, potentially leading to data-related issues.

Biessmann et al. [32] highlight bias in validation when the validation is mostly done by humans. This issue is especially prevalent with transparent ML methods [32]. While keeping humans in the loop is important in solving some data quality issues, it cannot be done to an unreasonable extent [32]. Similarly, [85] mentions manual steps in pipelines to cause data mistakes because of human involvement.

Choudhary [88] discusses data democratization of involving people without data knowledge into data pipeline processes. While the approach is popular and efficient for businesses, it can worsen data quality and reliability of the pipeline [88]. Unqualified people can introduce uninformed decision-making, developments and faulty data into the pipeline, all affecting data quality.

4.1.5 Data Loss

Data quality issues regarding data loss, data downtime, hidden data, and data leaks. These were mentioned 16 times (7%).

4.1.5.1 Data Loss

Data loss in its most direct nature was mentioned 9 times (4%). Data loss, and its adverse impact on both data completeness and overall data quality, was found to occur across the entirety of the data pipeline. During data extraction, the issues occurred due to various issues with data sources. In the transformation stage, the data was lost through third parties and general pipeline processes. The data was also lost due to its inherited nature to progress through stages. Table 12 presents a distribution of sources discussing data loss challenges for each stage.

Table 12 Distribution of Data Loss Challenges

Extract (4)	**Transform (3)**	**Load (0)**	**Entire Pipeline (2)**
[13], [18], [23], [39]	[18], [42], [46]		[5], [75]

a) Extract Stage

Munappy et al. [13] mention issues with data loss due to software failures, to cause data availability issues when extracting data for the pipeline. The loss of data is difficult to identify and using it in models can lead to underfitting.

Munappy et al. [18] describe data source failure as a problem when, for example, the data generation component is destroyed or even simply fails. Data generation failure that results in subsequent data loss can impact data completeness. Munappy et al. [18] also mention multiple other scenarios that can in many cases cause some form of data loss:

- inactive data sources which do not produce any data in the inactive state,
- authentication failure (e.g., expired credentials) restricts access to data resulting in breaks at the data collection step,
- data sending job failure results in data pipeline breakage.

Missing data is a common issue in solar panel sensor data that is related to hardware or software faults [23]. Software faults happen in data sources due to unreliable internet connection but also because of misconfigured software [23]. Hardware faults are simply identified as broken cables or sensors, yet they have data completeness ramifications of when a data is missing for longer periods of time [23].

Zhang et al. [39] recognize sensor malfunctioning, or device wearing non-compliance as a concern regarding raw data extraction and causing data loss within a data pipeline.

b) Transform Stage

A case study involving three companies [46], highlighted data loss happening during the process of aggregation, transformation, and cleaning, as a threat to data quality. There is also an instance where some encrypted data links sent for decryption are not returned, causing missing files and decrease in the final data product [46].

Pervaiz [42] highlights missing data as one of the challenges in data cleaning stage of the pipeline. Additionally, during data transformation, when data is modified, some of its parts are lost and it can inadvertently cause data quality issues through data loss [18].

c) Load Stage

There are no sources mentioning challenges related to data loss in the load stage.

d) Entire Pipeline

Somasundaram [75] mentions operating in distributed environments to cause data loss within the pipeline. Furthermore, data files can be lost during data transmission between nodes [5]. Data loss checks typically occur only at the end of the data pipeline, leading to potential issues with data completeness and overall data quality [5].

4.1.5.2 Data Downtime

Data downtime was mentioned 5 times (2%). Like data loss, data downtime was found to happen throughout the pipeline. Downtime presents a unique concern, yet in time-sensitive situations, its impact can mirror that of data loss. Downtime issues happened mostly due to latency, impacting the freshness and timeliness of moving data. Table 13 presents a distribution of sources discussing data downtime challenges for each stage.

Table 13 Distribution of Data Downtime Challenges

Extract (1)	**Transform (1)**	**Load (0)**	**Entire Pipeline (3)**
[19]	[36]		[44], [52], [75]

a) Extract Stage

Aaron [19] highlights data downtime as a concern, as incomplete data loading during analysis can result in inaccurate insights.

b) Transform Stage

Yaddow [36] mentions processing latency to introduce unexpected delays into real-time data monitoring and analytics performance.

c) Load Stage

There are no sources mentioning challenges related to data downtime in the load stage.

d) Entire Pipeline

Deshpande [52] strongly highlights data downtime as one of the more unsolvable problems within data pipelines. When data timeliness is key, data downtime and its inherited data loss can affect the pipeline decisions in ingestion and processing, or the final quality of the outgoing data.

Somasundaram [75] mentions distributed environments to cause data downtime. Osborn [44] identifies late data causing data freshness issues for downstream users.

4.1.5.3 Hidden Data and Data Leaks

Two unique challenges (1%) regarding the failure of important data to reach its intended destination were hidden data within data sources and data reaching unintended destinations due to leaks. Table 14 presents a distribution of sources discussing challenges related to hidden data and data leaks for each stage.

Table 14 Distribution of Hidden Data and Data Leaks Challenges

Extract (1)	**Transform (0)**	**Load (0)**	**Entire Pipeline (1)**
[19]			[89]

a) Extract Stage

Large organizations can have data silos with hidden or orphaned data [19]. This leaves the ingested data incomplete and leads to inaccuracies during analysis [19].

b) Transform Stage

There are no sources mentioning challenges related to hidden data or data leaks in the transform stage.

c) Load Stage

There are no sources mentioning challenges related to hidden data or data leaks in the load stage.

d) Entire Pipeline

Hilleary [89] elaborates on data pipeline leaks in cloud environments to cause downstream consequences. While not exactly data loss, it refers to data not reaching its intended destination and due to its improper handling, it introduces data quality issues.

4.1.6 Scalability

Data quality issues related to data pipeline scalability were mentioned 15 times (7%).

4.1.6.1 Technical Scalability

Technical scalability issues were mentioned 11 times (5%). Expanding the data pipeline with additional sources or increased complexity poses challenges to data quality assurance algorithms and techniques, leading to potential data loss or compromising the pipeline's ability to maintain data quality. Table 15 presents a distribution of sources discussing challenges related to technical scalability for each stage.

Table 15 Distribution of Technical Scalability Challenges

Extract (5)	**Transform (1)**	**Load (0)**	**Entire Pipeline (5)**
[33], [36], [37], [42], [65]	[41]		[27], [28], [75], [90], [91]

a) Extract Stage

When introducing new data sources, the pipeline needs to keep the complexity of data validation in check and points of variability should not increase [33]. This presents itself as a scalability challenge when new heterogeneous data sources are introduced into big data pipelines, and the quality of data needs to be maintained within the used normalization techniques [33].

Yaddow [36] discusses new data sources being added to the data pipeline as a threat to data quality through skewed analytics and reporting, causing false alarms and issues with the existing functionality. New data sources need immediate observability integrations to avoid them passing validation checks blindly [36]. Scaling can also cause downtimes and inconsistent data when new failure points are not detected by monitoring [36].

Pervaiz [42] investigated development data in low-resource environments and found data collection process over SMS failing to scale with information system formats evolving.

Biessmann et al. [37] mention a scalability issue with imputation methods within data pipelines. Common approaches to imputation are designed to work with numerical or categorical data, and it becomes problematic when datasets with millions of rows enter the pipeline [37].

Sadiq et al. [65] raise an issue where effectiveness and efficiency must be achieved during data management and integration to not waste critical resources.

b) Transform Stage

Huang et al. [41] highlight a challenge with workloads in hybrid-cloud environments in transformations. This becomes a scalability issue with data quality through label bias when acquiring labels from multiple sources, patterns and algorithms is restricted [41].

c) Load Stage

There are no sources mentioning challenges related to scalability in the load stage.

d) Entire Pipeline

Ensuring the quality of data in IoT applications is crucial, but it is a challenge because each application and its data pipeline has different requirements [90]. This becomes a problem as more applications are added, and each of them requires a data pipeline with quality assessment, leading to scalability issues [90]. An unsuggested solution by Byabazaire et al.

[90] is developing a different data quality assessment process for each application and integrating it into a single data pipeline. This would still leave the need to maintain multiple different data quality assessment processes [90].

Munappy et al. [27] mention a scalability requirement with big data, as the volume can vary over time. The pipeline must have a robust data quality assessment while having the capability to process real-time, high-volume structured, unstructured, and semi-structured data. For example, data quality monitoring system needs to be designed in a way that avoids bottlenecks [75].

Valencia Parra [28] highlights an issue with exhaustive algorithms when processing large volumes of heterogenous data. The algorithms are in place to help maintain high quality data, yet their complex nature causes increased search space, leading to scalability issues.

Merino et al. [91] mention the latency of data quality evaluations in data pipelines to prove costly. Data quality assessment built within the pipeline can have negative implications when the pipeline expands, and the quality framework must maintain its effectiveness [91].

4.1.6.2 Operational Scalability

Data pipeline operations restricting data quality during scaling were mentioned 4 times (2%). Operational scalability refers to the lack of adequate operational structure around and within the pipeline, hindering its ability to scale effectively while maintaining data quality assessment. Table 16 presents a distribution of sources discussing challenges related to operational scalability for each stage.

Table 16 Distribution of Operational Scalability Challenges

Extract (1)	**Transform (0)**	**Load (0)**	**Entire Pipeline (3)**
[47]			[32], [81], [84]

a) Extract Stage

Loetpipatwanich and Vichitthamaros [47] recognize a problem with managing many data sources with limited resources which is complicating adding data quality assurance to the data pipeline. The last decade has introduced many data quality management tools, but these products are often extensive, expensive, or difficult to custom [47]. However, this does not help fix the scalability issue for small businesses, researchers, and students for including a data assurance methodology in large data pipelines [47]. There are not enough frameworks and tools for maintaining data quality with limited resources.

b) Transform Stage

There are no sources mentioning challenges related to operational scalability in the transform stage.

c) Load Stage

There are no sources mentioning challenges related to raw data in the load stage.

d) Entire Pipeline

Biessmann et al. [32] determine scalability as one of the issues with using humans in data validation. Human audits are important, but additional automated techniques are needed for data validation to ensure room for development.

Writing data quality checks is not always a trivial task and it can be a difficult ask for data pipeline roles who are not data asset experts [81]. This becomes a scalability issue when building a pipeline of high data quality.

While [84] mentions Google's TensorFlow Data Validation (TFDV) [92] and Amazon's Deequ [93] to be effective data validation tools, it is significant manual work to write data validation rules for them. This becomes a scalability issue when the data feed is complex and there are not enough experts to assign for the task [84]. There are automatic constraint suggestions within Google TFDV and Amazon Deequ, yet this is still not considered scalable enough in demanding situations where data validation should rather be unsupervised and automatic [84].

4.1.7 Data Drift

Data drift was mentioned 14 times (6%). Data drift refers to gradual or sudden changes in the data, which can often not be identified without advanced measures.

4.1.7.1 Data Drift

Data drift poses a significant risk, as data drift may go unnoticed, bypass validation processes, and quietly degrade data quality. The problem may require resolution both within data sources and during data ingestion, as well as during data transformation. Table 17 presents a distribution of sources discussing data drift challenges for each stage.

Table 17 Distribution of Data Drift Challenges

Extract (7)	**Transform (6)**	**Load (0)**	**Entire Pipeline (1)**
[19], [30], [49], [68], [70], [81], [84]	[30], [36], [38], [41], [44], [83]		[32]

a) Extract Stage

Ryza [81] mentions spikes and anomalies as forms of data drift to cause a dangerous effect on data quality. For example, [49] highlights excessive data drifts to occur when sensor outputs a value much further from the true value.

Scott [68] mentions the natural enhancement and updates of source systems to influence data, causing bad data in the form of data drift. As the data can gradually change over time, it can cause data quality issues because the processes cannot automatically detect or adapt to the changes [70].

Inaccurate data introduced by unidentified data drift can cause instant data quality loss in the pipeline [19]. For example, the problem can silently arise from changing "en-us" to "en-US", which can then raise data quality issues within the pipeline [84].

Merino et al. [30] mention bias, seasonal data drift and clock-drifts as issues with data quality within data sources, and during data collection, preparation, and data analysis.

b) Transform Stage

Data distribution drift causes the model performance to degrade over time and it requires constant development of more complex algorithms in the pipeline [41].

Tu et al. [83] mention data drifts in recurring data pipelines to be one of the more indistinguishable problems affecting data quality. This can happen in the form of increasing nulls, change of units, change of value standards, and change of data volume [83].

Mattila [38] establishes changes in data, namely data drifts as a threat to data quality. Yaddow [36] mentions these anomalies in real-time data, and for them to lead to significant inaccuracies within the pipeline.

Osborn [44] identifies distribution errors and anomalous data points to be difficult to recognize by data pipelines. These issues can represent inaccurate data, but also plain data variety, which is why it needs to be solved carefully [44].

c) Load Stage

There are no sources mentioning challenges related to data drift in the load stage.

d) Entire Pipeline

Biessmann et al. [32] recognize statistical aggregation in datasets to reflect anomalies, resulting in data quality issues in pipelines.

4.1.8 Volume

Data quality issues caused by high or low volume of data were mentioned 11 times (5%).

4.1.8.1 High Volume

High volume issues were mentioned 9 times (4%). High volume of data can influence data integrity, and cause performance issues affecting data completeness through data loss. High data volume can also affect data quality measurements through latency issues, and data downtime. While closely related to both data loss and scalability issues, this challenge stands alone due to its specific nature. Table 18 presents a distribution of sources discussing challenges related to high volume for each stage.

Table 18 Distribution of High Data Volume Challenges

Extract (4)	**Transform (3)**	**Load (0)**	**Entire Pipeline (2)**
[20], [44], [46], [39]	[30], [36], [94]		[13], [75]

a) Extract Stage

Haro-Olmo et al. [20] signify a data quality problem with collecting large chunks of data with sensors in the field of IoT.

Zhang et al. [39] mention a core issue of unprecedented complexity and volumes of the digital data collected with sensor signals in dBM context. Overseeing that digital data is problematic because of the sheer amount of the 3-axial data points that are collected every day [39].

Munappy et al. [46] mention data collection agent of not handling huge volume of data, causing a problem with data completeness in the pipeline.

Osborn [44] mentions lack of volume monitoring in data pipelines causes overpopulation in models while also introducing data integrity loss.

b) Transform Stage

When discussing timeliness, [30] mentions too frequent and high-volume data to become an issue when sensors recalibrate too frequently. This causes data quality to diminish over time and causes problems during data analysis [30].

Minnaar [94] highlights problems with business intelligence systems and data standards in organizational data pipelines. When coupled with increasing data volumes and demand for optimal management, it sums up to a problem with maintaining data quality [94].

Yaddow [36] mentions increased data volume to cause delays in data processing and reporting, which can cause missed or incomplete data in the observable flow of the pipeline. Challenges with volume can also occur due to inefficient load balancing, causing bottlenecks to delay data for monitoring [36].

c) Load Stage

There are no sources mentioning challenges related to high volume in the load stage.

d) Entire Pipeline

Data flow instability can cause issues with continuous, intermittent or batch data [13]. When the inflow of data is too heavy, the pipeline may not be able to adjust properly [13]. This can cause problems in all stages, with the extraction and processing stages prone to fail with completing their operations on the data, and the loading stage by providing incomplete data to the source. This can compromise the dataset and therefore its quality.

Somasundaram [75] mentions high data volumes and rapid data velocity as a primary challenge in data monitoring within data pipelines. Delayed data validation with latency issues can lead to data inconsistencies and missed alerts [75]. Ensuring efficiency together with robustness and scalability is key [75].

4.1.8.2 Low Volume

Low volume issues were mentioned 2 times (1%). While low data volume does not influence performance, it still causes issues with data completeness and compromises the reliability of the whole dataset as it enters the pipeline. Low volume was mentioned to only be a problem near the extract stage. Table 19 presents a distribution of sources discussing challenges related to low volume for each stage.

Table 19 Distribution of Low Data Volume Challenges

Extract (2)	**Transform (0)**	**Load (0)**	**Entire Pipeline (0)**
[23], [44]			

Volume issues can arise in both extremes - there can be too much data, but lack of enough volume is causing data quality issues due to small sample size [44]. Without enough data, the dataset might not have enough *trustability* [44].

Pultier et al. [23] mention the small size of input data to be an issue. There are simply not enough years of data, and it can be considered unreliable in terms of data quality [23].

4.1.9 Compatibility

Data quality issues concerning incompatibilities between components surrounding or within the data pipeline were mentioned 9 times (4%).

4.1.9.1 System and Hardware Incompatibilities

Incompatibilities between systems and/or hardware pose a risk to data completeness, caused by data loss. The same incompatibilities can also simply disrupt data flow, leading to invalid data or degraded data quality. Table 20 presents a distribution of sources discussing challenges related to system and hardware incompatibilities for each stage.

Table 20 Distribution of System and Hardware Compatibility Challenges

Extract (2)	**Transform (2)**	**Load (1)**	**Entire Pipeline (4)**
[49], [53]	[36], [41]	[18]	[13], [42], [49], [60]

a) Extract Stage

Therrien et al. [49] highlight automated data collection to require different types of process instrumentation, equipment, and applications. As these parts need to communicate with each other, incompatibilities can result in data losses.

Unmaintained and mismanaged hardware used in data collection is another common source of errors and inconsistencies in data [49]. With wastewater sensors, the harsh environment fouls and degrades the sensing elements to the point where they do not produce accurate data [49].

Taverna et al. [53] discuss data format variations and compatibility issues to cause difficulties with geothermal and geospatial data. This complicates adding new data sources to the pipeline, because the formats have many variations that are not compatible with the designed pipeline [53].

b) Transform Stage

Huang et al. [41] identify system performance drifts as a concern impacting the accuracy of the data. Updates and changes in the infrastructure software or its underlying hardware result in workload changes, modifying performance data that goes through the pipeline [41].

Yaddow [36] mentions data processing issues related to incompatibilities with downstream systems to result in data loss and errors.

c) Load Stage

Munappy et al. [18] claim that data parsers in data pipelines become incompatible or obsolete, the resulting sent data is invalid. This is known as dirty data, and it mostly happens during data transmission between nodes [18].

d) Entire Pipeline

Picking a less-optimal data management platform can complicate data analysis and create data debt problems [60]. This is a problem of flexibility in the long run [60]. Munappy et al. [13] mention a similar issue with when a data silo is of poor architecture or uses legacy systems or outdated company culture, the data can become isolated without reaching its target, and therefore causing incompleteness in data.

Pervaiz [42] recognizes a legacy data issue to be unavoidable with development data in low-resource environments. Data collection process over SMS simply cannot cater to new requirements [42].

Therrien et al. [49] highlight a general problem with simple databases not being able to handle complex big data.

4.1.10 Logical Errors

Data quality issues that were caused by rare errors in the overall design and logic were mentioned 7 times (3%).

4.1.10.1 Logical Errors

This section unites all the data quality issues that were caused by logical errors, meaning misinterpretations, lack of synchronization and access, or other faults caused by uncoordinated processes. These issues usually have a clear solution, yet they are still found occurring in data pipelines. Table 21 presents a distribution of sources discussing challenges related to logical errors for each stage.

Table 21 Distribution of Logical Error Challenges

Extract (4)	**Transform (2)**	**Load (0)**	**Entire Pipeline (1)**
[18], [23], [46], [49]	[5], [18]		[49]

a) Extract Stage

Munappy et al. [18] describe unexpected data as a typical fault causing partial failures in data pipelines. When a new source is added without notification the next stage of the pipeline might fail [18]. This is primarily a logical error but can also be considered as data loss or even a scalability issue if the logical error is solved in the subsequent data pipeline stages by disregarding the added source. Munappy et al. [18] expand on the issue claiming that adding a new source can generate data different from the existing sources, therefore causing data ingestion failure.

Pultier et al. [23] bring out an issue with solar panel sensor data where data sources have different sampling rates, with clocks that are not synchronized tightly. The synchronization error together with time zone issues and data delays, aggregate to a clear data quality issue [23].

Three companies participating in a data pipeline case study [46], mention a problem with data collection agent not having access to entire data.

Therrien et al. [49] mention that collecting data without a clear motivation or purpose may result in the creation of data graveyards. These vast repositories of data are unlikely to be utilized and contribute to noise within the data collection.

b) Transform Stage

When logical changes cause data drifts and change in data distribution, the data pipeline will sometimes fail because of existing incompatible logic [5]. The essence of the problem is the aging of the data pipeline when it is not updated constantly [5].

Munappy et al. [18] bring out a prominent issue with unclear definitions and misinterpretations in data transformation. Data quality decreases with every wrongly interpreted value used for analysis.

c) Load Stage

There are no sources mentioning challenges related to logical errors in the load stage.

d) Entire Pipeline

[49] mentions data stored in multiple storage sites, resulting in problems with access.

4.1.11 Security

Data quality issues that happened due to authorization and security faults were mentioned 6 times (3%).

4.1.11.1 Unauthorized Access

When data moving through the pipeline can be accessed freely, it immediately causes concerns with its integrity, and there are no assurances if it has been modified by only authorized roles. Even when not modified, data breaches are confidentiality issues, which can in turn, immediately reduce the quality of the data. Unauthorized access issues were mentioned only within the context of the entire pipeline. Table 22 presents a distribution of sources discussing challenges related to unauthorized access for each stage.

Table 22 Distribution of Unauthorized Access Challenges

Extract (0)	**Transform (0)**	**Load (0)**	**Entire Pipeline (6)**
			[13], [46], [57], [60], [75], [95]

Catarci and Scannapieco [95] highlight a significant problem in the data pipeline from a security and data quality perspective. The increasing availability of data for policymaking decisions brings forth challenges of confidentiality and integrity. Large-scale privacy breaches and potential manipulations by unfaithful data operators can disrupt policymaking processes, particularly in big data systems. The paradigm shift to electronic voting systems introduces vulnerabilities that, if exploited, can mislead decisions on critical issues, jeopardizing the essence of modern democracies. This underscores the critical need to address security concerns in the early stages of the data pipeline to ensure the integrity and confidentiality of data for informed policymaking. [95]

Sacolick [60] suggests data debt to cause security and trust problems if left unattended through the data pipeline. Munappy et al. [13] also mention data quality challenges with data governance and security while dealing with real-time data.

Somasundaram [75] highlights keeping data secure and private to be a key factor in monitoring. Sensitive data needs to be handled accordingly, as breaches can compromise data integrity [75]. Similarly, Munappy et al. [46] mention the risk of an unauthorized person having access to the data. Without safeguards against external attacks, the integrity of the data is at risk.

Byabazaire et al. [57] highlight a specific security and trust concern regarding data travel through data pipelines. The open-source nature of IoT has a big security concern, and there is no guarantee that the data will retain its quality in data pipelines [57]. Data sources can advertise data quality, but there is no concrete security regarding the quality measure when data is shared or modified by another entity [57].

4.2 Solutions

This section handles the challenges in an analogous manner as they were classified in 4.1. One key difference is that the solutions and mitigation strategies are often more general and are often targeting all the second level challenges under the main eleven groups. The solutions often have a more significant effect, and they are effective on a larger scale, which complicated the process of assigning the solution to a specific second level challenge. This is further amplified by some strategies (e.g., pipeline architecture improvements) being very efficient against all data pipeline challenges. However, the solutions still retain the structure used in 4.1 to a high degree to provide an efficient overview and enable better readability.

Figure 9 provides an overview of the proposed solutions, with some level two categories being handled together (e.g., Raw Data and Data Untrustworthiness). The figure includes the main categories of solutions for each issue. Some challenges include more general sections due to the distinct nature of each solution or lack of solutions hindering classification.

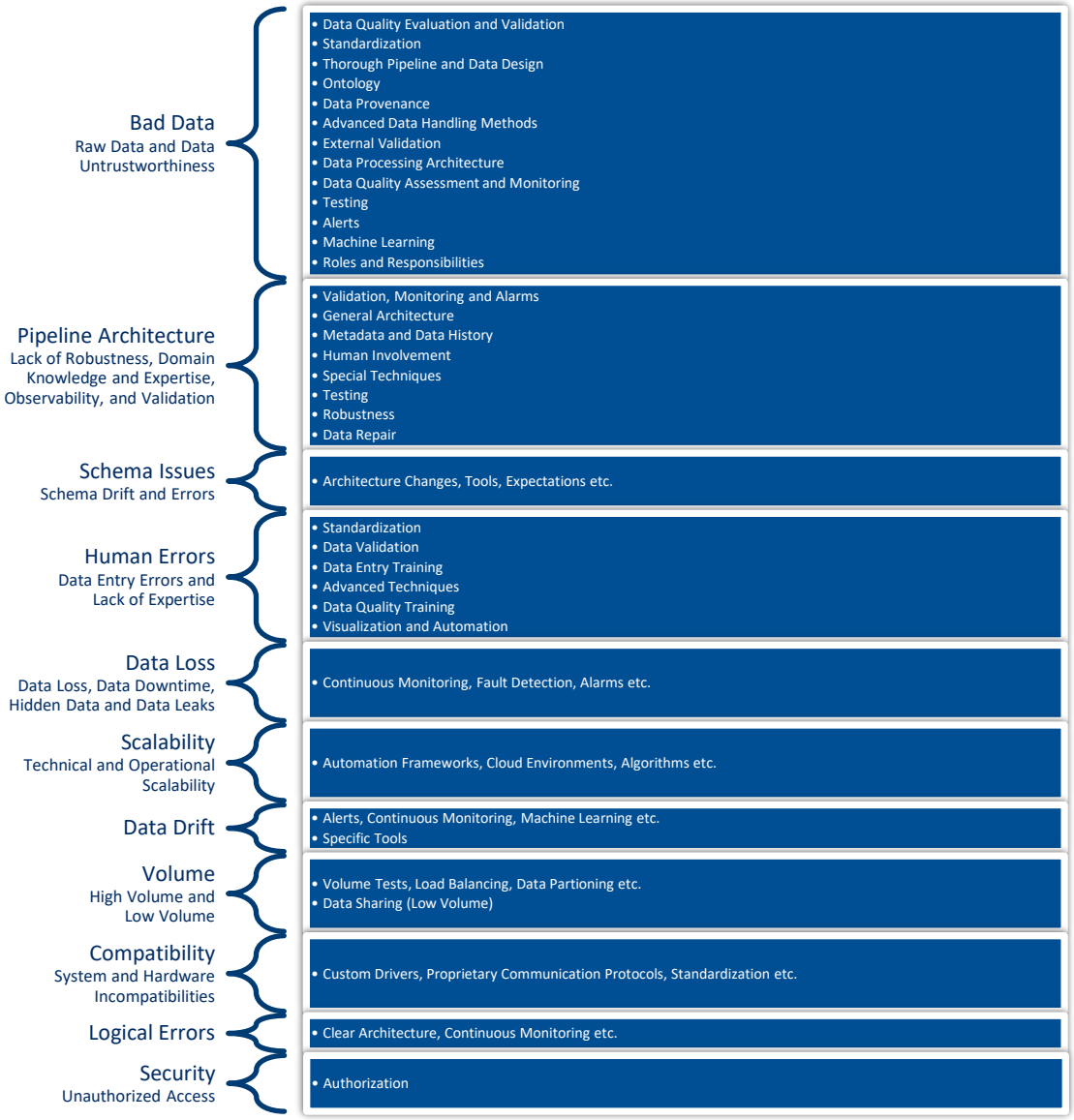

Figure 9 Solutions Overview

4.2.1 Bad Data

Bad data had two main branches of problems related to their heterogeneous nature and lack of trustworthiness. In total, 48 sources discussed 67 solutions.

4.2.1.1 Raw Data and Data Untrustworthiness

The solutions are often similar in their nature, making it challenging to separate the solutions into two branches. However, the solutions are informatively classified into 13 general groups, while also considering the stage of the pipeline. Each group pursues its distinct agenda aimed at addressing the inherent complexity of raw data and enhancing or guiding decisions based on the trustworthiness of data. Table 23 presents an overview of categorized solutions, and their respective stages, sources, and number of mentions.

Table 23 Bad Data Solutions

Solution	**Stage**	**Sources**	**Mentions**
Data Quality Evaluation and Validation	Extract	[19], [20], [23], [28], [32], [38], [47], [56], [61], [68], [70], [94], [96]	13
Standardization	Extract	[18], [19], [25], [26], [30], [42], [53], [54], [97]	9
Thorough Pipeline and Data Design	Extract	[19], [24], [26], [33], [48]	5
Ontology	Extract	[26], [30], [50], [95]	4
Data Provenance	Extract	[55], [60]	2
Advanced Data Handling Methods	Transform	[19], [25], [30], [31], [37], [46], [64]	7
External Validation	Transform	[41], [46], [54], [62]	4
Data Processing Architecture	Transform	[26], [64], [66]	3
Data Lineage	Transform	[29], [60], [85]	3
Data Quality Assessment and Monitoring	Entire Pipeline	[35], [45], [46], [68], [83]	5
Testing	Entire Pipeline	[33], [43], [44], [69], [98]	5
Alerts	Entire Pipeline	[18], [27], [52], [58]	4
Machine Learning	Entire Pipeline	[35], [36]	2

Roles and Responsibilities	Entire Pipeline	[97]	1

a) Extract Stage

The solutions and mitigation strategies against bad data within the extract stage are classified into five groups: Data Quality Evaluation and Validation, Standardization, Thorough Pipeline and Data Design, Ontology, and Data Provenance.

Data Quality Evaluation and Validation

In combatting the diversity and heterogeneity of IoT sensor data, Haro-Olmo et al. [20] propose a customizable decision model based on the Decision Model and Notation (DMN) [99]. The model evaluates completeness and accuracy of the initial data to classify the data as adequate, sufficient, poor, or unusable [20]. Introducing a model to evaluate and label diverse incoming data provides the user with options to assess the data and make decisions to benefit quality of the loaded data. Similar models can be developed in other larger or extremely specific fields, and the data pipelines can take advantage of them to improve incoming data quality.

Sakdas: A Python Package for Data Profiling and Data Quality Auditing [47] is introduced as a package[3] to help understand the data using statistical and visual approaches, and data auditing. Regarding raw data challenges, Sakdas can help profile and audit data to help in the development of data quality assessment mechanics by pinpointing data quality weaknesses and strengths. Sakdas uses data completeness, integrity, and consistency as the three statistical dimensions in its data assessment [47]. These dimensions are all integral parts of the full definition of data quality.

Zhang et al. [39] propose a data quality analysis pipeline to assess the data quality of the sensor data for dBM research. Data sanity checks against device specifications will provide several valid data points that are sensible to be used for processing. Sanity check includes metrics of sampling frequency, valid range, and invalid value/error code. After some of the data is removed, the challenging detection of wear or non-wear is analysed as the data comes from a wearable device. Zhang et al. [39] then propose an extensive quality model for the data to pass through and derives a compliance report. The quality model and compliance report provide qualitative measures to the data for determining whether the sensor data is useful for analysis. [39]

In the realm of big data, a traditional approach involves leveraging larger datasets, treating unneeded parts of raw data as mere noise that can be disregarded [23]. This solution enables increased data quality through data availability, at the expense of using more resources. However, [23] detects data quality changes in raw data using fault detection algorithms and careful data analysis. In other cases, [23] opts to drop the data entirely in the pursuit of higher overall data quality if the data is significantly different.

Marketing Evolution's Marketing Measurement and Attribution Platform targets marketers with data pipelines to address the issue of substandard quality marketing data [61]. The platform verifies data quality by analysing if the data has arrived in the correct format and in a timely manner, checks that the data is corresponding to its intended influence, and finds if the data has insightful values about the customer journey [61].

[3] https://pypi.org/project/sakdas/

Valencia Parra authored a comprehensive doctoral thesis on enhancing big data pipelines through data preparation, data quality, and the distribution of optimization problems [28]. Regarding data quality within a data pipeline, [28] proposed a hierarchical structure of business rules to validate data attributes, measure data quality dimensions, assess the data quality, and produce recommendations on the usability of the data. The ability to generate usability decisions and repair options based on usability profiles provides an extraordinary opportunity to minimize data quality issues within a data pipeline [28].

Bhardwaj [96] proposes a measurement framework for big data (MEGA) where data issues are identified and analysed before they reach high-impact decision-making processes. The main measurable validity attributes in the framework include accuracy, credibility, and compliance [96]. MEGA runs in parallel with big data pipelines and monitors data for data quality issues [96].

Contributions from Database Management Systems research on data profiling literature [100] are helpful in solving data correctness and data consistency challenges. Using statistical techniques in data validation is becoming increasingly researched. There are also solutions of rule-based systems like trifecta (now alteryx[4]) OpenRefine[5], TensorFlow [92] and Deequ [93]. The validation can be set up with parameters from training data, which enables partially automatic data type constraint generation. [32]

Redyuk et al. [70] compare the results of their automatic data quality validation solution with Deequ [93], TensorFlow [92] and statistical testing. All these approaches and tools can help detect erroneous data and provide much needed validation to ensure data quality during data ingestion [70].

For solving erroneous data entering the pipeline, [68] signifies the importance of adding input profiling, validation, and data error management. Data error management can be a combination of people, processes, and tools, while enforcing a standard format for error reporting [68]. Another key approach is to identify faulty processes and data issues as close to the source as possible, providing most safety against issues that have corrupted the data with snowballing [68].

Aaron [19] advises implementing monitoring mechanisms during data ingestion to confirm data synchronization; any failures should trigger immediate notification to data engineers. This can alert data engineers to act when there is an erroneous data record. Proactive data validation is key in finding recurring data problems early [19]. When data is validated early during data ingestion, it can pinpoint issues with source data, and therefore enable changes in processes to accommodate to the pattern.

Huang [56] signifies data cleaning, data validation, and data quality control as ways to find inaccurate and unreliable data, improving the overall qualitative performance of the pipeline. Data quality control can be achieved through attaching accuracy and completeness metrics to the data, allowing efficient monitoring throughout the data's journey [53].

Mattila [38] created a process model for incorporating data quality validation into pipelines. Although validation is important for measurements and can help identify specific problems with raw data, it does not improve the data quality by itself and needs to be combined with other solutions [38].

Minnaar [94] proposed a framework for identifying data quality issues, to assist decision-making within a data pipeline. The model focuses on assigning tags to data items, with main

[4] https://www.alteryx.com/about-us
[5] https://openrefine.org/

attributes being data type and data scale. Data scale indicates whether the data item is a text, nominal, dichotomous, ordinal, interval, discrete, continuous, percentage, ratio, or currency value. Data type refers to common classification of string, integer, float, date, date-time and boolean. Moreover, data tags are linked with a record (with relation to storage), a source (with relation to the asset and users) and an indicator (with relation to reports). The model, along with a proposed tool to utilize the framework in building data pipelines, provides opportunities to find red flags in raw data, e.g., duplication, incompatible stores and manually collected business critical data. [94]

Standardization

Pervaiz [42] describes Cold Chain Information System (CCIS) as a form of standardization to deal with raw data regarding health systems in low-resource environments. Pervaiz [42] solves raw data issues by providing availability of information, data models, data reporting and an overall processing system for health organizations to use. Data reporting should be supported by engaging managerial responsibilities to achieve good report rates, and therefore good data quality [42]. For SMS reporting, CCIS enables a data entry wheel to form nine-digit reports and provide error detection through checksum [42].

Regarding various standardized formats in source data, [25] mentions organizations to use custom scripts for data conversion. Voropaeva [97] mentions standardization of data models and documentation as a way of improving not only data quality, but quality in general.

Ilyas and Naumann [26] suggest standardization as a needed solution to deal with various data sources, and their produced raw data. Standardized data and meta-data can result in simpler data provenance models, but it would provide the possibility for the data pipeline to maintain high data quality [26].

Aaron [19] suggests standardization to solve issues with conflicting formats and overall inconsistencies. Proper data governance with data policies, guidelines and standards can function as a source of data *trustability* within the pipeline [19].

As a solution to issues related to missing metadata, [54] finds solutions in standardization and building automatic concept annotators. Similarly, Merino et al. [30] suggest standardization and semantic annotation to improve data quality early.

Mitigation strategies against mismatching data types and incorrectly formatted data include standardization of data formats, data conversion and defining a new data extraction method able to scrape data in all different data formats [18]. Standardization helps with enforcing correct data types in source data and provides the ability to design a standardized data pipeline. Data conversion helps with converting data not matching its type (either in a similar type or a mistake) to usable form for the data pipeline. Defining a complete data extraction method that expects and knows all different data formats is an ideal solution [18].

Taverna et al. [53] mention standardization to contribute towards data reusability. Consistent formatting can reduce preprocessing efforts and offer insurance of data quality within the data pipeline [53]. Geothermal data standards introduced in [53] facilitate high-quality data analysis within data pipelines.

Thorough Pipeline and Data Design

Rahman et al. [48] try to understand complicated and unreliable traffic data sources and provide insights on understanding the data and the capacity to model correlations for potential further work and produces a data pipeline while doing so. Rahman et al. [48] fortify the idea of conducting thorough research and modelling on complicated raw data to help design data pipelines that maintain high data quality.

For overcoming the data quality issue with heterogeneous data in any IoT context, [24] proposed an IoT big data pipeline architecture. The architecture takes advantage of DMN4DQ [101] approach to enable measurement and evaluation of sensor data quality [24]. Together with DMN [99], the architecture allows the automatic measurement and assessment of the usability of the data, providing options to improve data quality with the correct extraction and selection of data [24].

In discussing raw data quality issues, [26] suggests that often the only solution is to solve the issues within the data sources. Improving the quality of the data within the data pipeline can be a futile effort and communication with data owners or creators is often the only choice [26].

As a solution to record linkage and noisy data in general, [33] insists on defining fault-tolerant evaluation in pipelines. Whiles not a fully defined solution, [33] considers it important to have this step and using various techniques ranging from probabilistic to up-front methods to actively find and solve data quality issues before the data moves downstream.

Aaron [19] highlights the importance of implementing robust scans and data matching algorithms in data ingestion, to find and merge duplicate records. This can also be done with a separate data integration pipeline [19].

Ontology

Catarci and Scannapieco [95] claim consistency and accuracy to be the most important aspects in data quality assessment, with confidentiality also getting mentioned. Consistency is often assessed by checking if data follows rules for integrity. However, [95] promotes Ontology-Based Data Access (OBDA) which defines rules from ontology, providing unique, validated standards for the entire system. Automated techniques in OBDA systems leverage inference capabilities to identify and rank inconsistencies, sparing the need for laborious checks on diverse sources. Addressing data accuracy, OBDA introduces a novel approach by enriching the specification with statements defining expected accuracy levels. This allows OBDA to distinguish between the world's knowledge (ontology) and the data's knowledge, providing insights into accuracy issues. Managing incomplete data, OBDA specifications set acceptable accuracy standards, triggering specialized algorithms for assessment. [95]

Abdellaoui et al. [50] use an ontology-based solution to design the data integration system schema and extends it by adding data quality rules to it. Merino et al. [30] suggest using ontology and knowledge-based systems in reducing uncertainty and issues with missing data, helping maintain data accuracy. Ilyas and Naumann [26] also mention a controlled vocabulary or a repairing ontology to help in solving issues with data.

Data Provenance

When data is incomplete or lacks proper metadata, there is an option of building a custom solution for data quality assessment [55]. The key activities include tracing lineage to achieve dependencies, describing internal and external characteristics of data, and recording the interaction of data with the outside world [55]. Defining and collecting new metrics, metadata, lineage, and logs from the incoming data can help reduce the incompleteness of it [55].

Sacolick [60] suggests a shift-left quality assurance practice, to deal with data quality issues and data debt as early as possible. Data catalogues, data linage tools, and metadata management systems are all suggested to help reduce the risk of data debt [60]. Data

governance through roles and metrics can help maintain data models and provide data quality assurances [60].

b) Transform Stage

The solutions and mitigation strategies against bad data within the extract stage are classified into four groups: Advanced Data Handling Methods, External Validation, Data Processing Architecture, and Data Lineage.

Advanced Data Handling Methods

Data imputation is a common solution for solving missing values, but it often focuses on numerical or categorical data [37]. Data imputation solves missing values by filling the gaps, and therefore increasing data quality and reducing data debt [37]. Biessman et al. [37] introduce DataWig, that is a missing value imputation approach that works efficiently with heterogeneous data types and unstructured text.

In countering missing values from heterogeneous data, [31] conducted benchmarks for various imputation methods to increase data quality. Jäger et al. [31] found imputation after data ingestion to increase the final model performance 10-20% in 75% of their experiments. Simpler imputation methods yielded competitive results, while random-forest-based imputation achieved the best improvements in data quality [31].

When cleaning and transforming data of inferior quality, [25] suggests using tools that use pattern detection and heuristics. Wrangler [102] and Potter's wheel [103] is a set of tools that show users the transformations as they progress, with the goal of engaging users in the cleaning process [25]. For duplicate data, [25] found defining similarity functions to be effective in finding clusters of matching entities. Common attributes and probabilistic record linkage are some more recommended solutions to find and deal with duplicate records [25].

Merino et al. [30] mention false data detection and domain-specific algorithms to be crucial during data pre-processing. Some examples include cross-validation mechanism can improve crowd-sensing data quality, correlation method and principal component analysis in sensor networks, outlier detection for intelligent car data analytics, and missing value prediction for traffic status data.

Livera et al. [64] handle outliers with manual approaches, imposing physical limitations, visual inspection of scatter plots, and applying variation limits between data points methods and statistical and comparative tests (e.g., Sigma rule or standard boxplot rule).

Merino et al. [30] list many different cleaning methods for solving for solving problems related to outliers, noise, uncertainty, and missing data throughout the data pipeline. For example, interpolation based on principal component analysis (PCA) can estimate missing data [30]. Solutions can vary, and [30] mentions PCA, Partial Least Squares (PLS), Kalman filter, Scheffe test to be efficient in cleaning outliers; PCA, PLS, Kalman filter, Empirical mode decomposition, Savitzky-Golay filter and multivariate thresholding to solve noise-related problems, probabilistic imputation to help with uncertainty in data; and PCA, PLS, association rule mining, clustering, k-Nearest Neighbor, Singular value decomposition, Clustering and probabilistic matrix factorization to be efficient against missing data. The solution is often very situation-dependent, but [30] is helpful in classifying data quality issues together with the best possible cleaning methods.

To address a unique challenge posed by different regions, languages, and units, [19] recommends developing models to handle regional inconsistencies, a practice also referred to as internationalization.

Munappy et al. [46] mention the changing of data type as a form of automated solution. Parsing errors are easily identifiable, and in certain instances, they may originate from data type issues, allowing for potential automatic corrections. Similarly, [19] suggests data type transformation as a form of mitigation against incorrectly formatted data. This is a kind of pre-processing approach that can unify the incoming data for aggregation.

External Validation

Immonen et al. [62] insist on the need for companies (data sources), to ensure the quality, trustworthiness and reliability of the data used and collected in their business. Organizational policies as a form of standardization that both the company and data pipeline processing stages can follow can help ensure data quality [62]. Each policy can be applied for different purposes of data collection [62]. Using the policy, data processing can select the datasets, quality attributes and metrics for the quality assessment [62].

Huang et al. [41], motivated by Intelligent Alerting system built in TellTale at Netflix [104], developed a Slack-based labelling tool that post summaries with different links and allow engineers to label data points by reacting with thumbs-up or thumbs-down. This also allows studying the distribution patterns of the annotators and subjective anomalies will be solved with majority vote or dropping the label [41]. After observing the labelling inconsistency, [41] also introduced standardization to the labelling process. Slack-based tool combined with standardization greatly improved the quality of the data points [41].

Munappy et al. [46] discuss data processing of an automotive company and its data quality analysis pipeline. The company has implemented quality assurance, and sends quality reports to data sources, for them to initiate investigations and fix quality issues with their data [46].

Rahman et al. [54] claim that involving domain experts in data validation can help with catching errors, as opposed to data engineers who are unfamiliar with the domain. Domain experts can validate data integration datasets, identify, and fix errors, augment missing data, restructure the data, interact with summaries and outliers, and ensure explainability within the models [54].

Data Processing Architecture

Ilyas and Naumann [26] state that when working with raw data, it is important to model both processes and data generators within the system, rather than solely focusing on data and errors. Additionally, data errors without context and erroneous data processes need to be identified. It is also essential to extend the concept of provenance to encompass potentially faulty processes, internal computation, and data generation steps. The design of algorithms and systems capable of efficiently tracing this extended provenance is needed, along with developing repair operations for errors and processes that go beyond mere data deletion or replacement, which is the current norm. [26]

Livera et al. [64] provide a data processing methodology for bad photovoltaic system data. The process starts with a consistency examination, looking for gaps, and repetitive and duplicate records. The next step is data filtering, to restrict measurements to only certain times. After filtration, the pipeline moves on to identification of invalid values by searching for null values, applying limits to measurements, and applying statistical or comparative methods. Then the pipeline finds missing data rate and executes data deletion to either get rid of unmanageable data or data interference techniques to fix the data. After this, the pipeline aggregates the data into daily, weekly, monthly, or annual values, and the data quality processing is complete. [64]

May and Fuller [66] mention the use data reports within data pipelines to have the option of validating ranges, extremes, and overall data quality from time to time.

Data Lineage

When data becomes bad during transformation, it is important that the pipeline has data lineage [85]. This provides the opportunity to understand which row caused the data error as it works as a kind of debugging feature in data [85]. The importance of data provenance is also mentioned in [29] as it can assist working with inaccurate data.

Armbrust and Lappas [85] offer a solution of out of order data created in the pipeline with reconciliation. The pipeline should have a reconciliation view and backing table with extra rows and columns to enable the possibility to amend erroneous data [85]. When data becomes bad, automated processes or responsible roles can look back on the original form of the data for potential solutions.

Data lineage is an important property to introduce and observe, as it represents the state and status of the data across its life cycle and enables therefore solutions to improve on the data quality [60].

c) Load Stage

There are no sources targeting the load stage in solving challenges related to raw data or data untrustworthiness.

d) Entire Pipeline

The solutions and mitigation strategies against bad data in the entire pipeline are classified into five groups: Data Quality Assessment and Monitoring, Testing, Alerts, Machine Learning, and Roles and Responsibilities.

Data Quality Assessment and Monitoring

To combat the usage of poor-quality data in big data pipelines, Zou and Xiang [45] propose a novel rigorous big data quality measurement architecture to help provide much-needed big data quality assurance through all stages of the pipeline. The architecture focuses on six dimensions (Volume, Variety, Velocity, Veracity, Validity and Vincularity) to measure throughout the data pipeline phases of data extraction, data loading and preprocessing, data processing, data analysis, and data loading and transformation. Initial results did confirm improved data quality after implementing the model and using it to modify defective data elements. [45]

Data validation is an important part in avoiding or identifying erroneous data and the resulting performance degradation in data pipelines [46]. Research and practice suggest automated data validation in data pipelines to allow traceability to provide opportunities to counter bad data [46]. The countermeasures come in the form of developing manual and automated, or semi-automated mitigation strategies [46]. This includes functional changes (e.g., changing data type) or sending alarms [46].

In developing Auto-Validate-by-History, Tu et al. [83] provide data pipelines with a framework that can detect inconsistencies in data based on previous data. When working with raw data, the framework provides an opportunity to verify data against previously processed data. Although the framework needs previously verified data as a form of history as input, its output is a great detector for when the incoming data is of varying quality and needs to be addressed.

Scott [68] signifies adding data validation in the sense of each data record being assessed based on business rules in each stage of the pipeline. This is a basic data quality control measure of whenever a record fails validation, it provides an actionable error message, while the processing continues with good records. Adding profiling can detect questionable data from entering the pipeline or moving within the pipeline. This statistical analysis can happen in the form of holistic assessment on the whole dataset, to prevent processing some data or to identify suspicious data and data decay. Examples of this analysis can provide most frequent patterns, unique counters, count of nulls, or rate of convertible to number. [68]

Wallace [35] suggests implementing data profiling tools and anomaly detection algorithms to identify bad data. Automation is also integral, as it allows teams to monitor the data more efficiently and find problems early [35].

Testing

Yaddow [69] highlights the popular approach of testing data pipeline workflows to ensure data quality. Data entering the pipeline is often untrusted, and tests provide a qualitative measure of whether the data is valid [69]. Tests should be linked with various data quality characteristics, and they provide insight to decision-making during data preparation and cleaning [69]. Yaddow provides numerous examples of tests in [69] for assessing each distinct dimension of data quality, including accuracy, completeness, conformity, consistency, integrity, precision, timeliness, and uniqueness. Accuracy tests can range from simple null checks to comparing values with pre-set requirements, and similar variety applies to all dimensions [69].

Dazzeo [43] tried tackling the bad data issue with fault injection to identify weak points in the data pipeline. This approach is commonly used in relation to ensure data quality and it allows to introduce uncertainty into the data and its effect on the pipeline [43]. The ambiguity and errors generated by the faults allows a better understanding of the pipeline and its responses to faulty data [43]. Perturbations can help users to then identify sensitive parts of the data and optimize the pipeline accordingly [43]. The fault injection techniques used in [43] were effective in the used experimental data, but it is challenging to identify all sources of uncertainty with real-world data.

Nawaz [98] covers data quality testing in data pipelines with Great Expectations Python library[6]. The library enables data testing related to table shape, missing values, unique values, data types, sets and ranges, string matching, datetime and JSON parsing, aggregate functions, multi-columns, distributional functions, and file data assets [98].

Ardagna et al. [33] discuss the importance of monitoring and testing within big data pipelines. Verifying data provenance is one of the key factors in assessing the accuracy and overall quality of the data [33].

Osborn [44] suggests taking advantage of dbt's [105] not null and unique tests to solve issues regarding missing values and duplicate data. Relationship tests in dbt can monitor issues with referential integrity and help discover altered or deleted data items [44].

Alerts

Kauhanen [58] designed data validation for railway operation data, with defining a rule store and rule processor that can be used to combat the heterogeneous nature of railway data. The rule processor facilitated the utilization of separate validation functions for various data streams, and an alerting system was implemented as a form of data quality monitoring [58].

[6] https://github.com/great-expectations/great_expectations

Munappy et al. [27] insist on data pipelines to be built with a strict monitoring mechanism to continuously monitor the quality of the data. When problems arise, an alarm can automatically be sent to the concerned department to either fix faulty sensors or take other actions [27].

Deshpande [52] introduces AWS Glue[7] to handle bad data entering the data pipeline. AWS Glue allows setting up data quality rules and alarms, while also allowing analysis and categorization of data in data pipelines [52].

Munappy et al. [18] suggest sending out alarms if the format is changed in all stages of data pipeline starting from data ingestion. Companies are also known for using versioning for data formats as a form of silent alarm for data pipeline teams to adjust their data pipelines when data format has been changed [18].

Machine Learning

Yaddow [36] suggests implementing advanced data quality checks to identify and work with poor-quality data, resulting in improved data integrity. Advanced data quality checks are characterized by their efficient resource utilization, ability to identify both false positives and false negatives, ease of maintenance, and domain-specific approach [36]. Finding false positives and false negatives can be achieved through feedback loops and historical data analysis [36]. Domain expertise can be achieved through AI and machine learning, which can learn to write data quality checks just based on the data [36].

Wallace [35] mentions data teams to take advantage of machine learning when automating data classification, predictive analytics, and anomaly detection techniques. Machine learning models and algorithms can be used to extract additional quality metrics and patterns from data, and therefore determine the quality of the data [35].

Roles and Responsibilities

When proposing solutions in a case study, [97] suggests defining a completely new role of data warehouse quality engineering manager. The responsibilities include maintaining data assets, standardizing data models and attributes, maintaining metadata repository, monitoring, and reporting data quality, root cause analysis and general overseeing of the data pipeline life cycle [97]. In many cases, the new supporting near-data role is necessary to help enforce the many solutions proposed against raw data issues, thus improving data quality within a pipeline.

4.2.2 Pipeline Architecture

The pipeline architecture challenges revolved around lack of four key elements: robustness, domain knowledge and expertise, observability, and validation.

4.2.2.1 General Pipeline Architecture

The sources discussed solutions during processing exactly but also more generally in the entire pipeline. The solutions are also formatted in 8 categories and instead of following the four key elements because of the overlapping nature of the challenges and solutions, and the ability to provide more value through broader categorization. In total, 43 sources offered 55 solutions to improve the architecture of the pipeline to facilitate better data quality. Table 24 presents an overview of categorized solutions, and their respective stages, sources, and number of mentions.

[7] https://docs.aws.amazon.com/glue/latest/dg/what-is-glue.html

Table 24 Pipeline Architecture Solutions

Solution	Stage	Sources	Mentions
Validation, Monitoring and Alarms	Transform	[13], [22], [24], [106]	14
	Entire Pipeline	[13], [27], [35], [44], [51], [73], [78], [80], [81], [107]	
General Architecture	Transform	[18], [49], [71]	8
	Load	[33]	
	Entire Pipeline	[36], [53], [66], [108]	
Metadata and Data History	Transform	[13], [35], [49], [88]	8
	Entire Pipeline	[97], [107], [109], [110]	
Human Involvement	Transform	[50], [55], [111]	7
	Entire Pipeline	[32], [75], [97], [112]	
Special Techniques	Transform	[25], [71], [74], [75], [76]	5
Testing	Entire Pipeline	[43], [67], [81], [88]	4
Robustness	Load	[113]	4
	Entire Pipeline	[46], [74], [79]	
Data Repair	Transform	[49], [82], [95], [114]	4

a) Extract Stage

There are no sources targeting the extract stage in solving challenges related to pipeline architecture.

b) Transform Stage

Sources that tackled the four key architectural challenges during the transformational phase of the pipeline found it integral to implement validation strategies, conduct data repair, take advantage of data history, and involve humans in processes. In other cases, specific data handling techniques and general architectural ideas were suggested to help with data quality.

Validation

Because data pipelines have a high impact on data quality, strategies for data validation should be adapted to the data pipeline architecture [22]. Furthermore, the stage-based results (e.g., data smells and errors) of the validation should then be investigated and improved upon [22]. Detection tools can reveal data smells and send alerts of ineffectively processed data [22].

Haro-Olmo et al. [24] reference DMN4DQ [101], which is a methodology that allows for the automated creation of suggestions regarding the probable usability of data based on its quality level. DMN4DQ enables the validation of data attributes, the quantification of data quality aspects, and the evaluation of overall data quality levels [24]. The hierarchy of business rules that enables all this, is supported by decision model and notation paradigm (DMN) [24, 99]. Determining the usability of data can improve the decision-making of the functions in processing data with data of higher quality more extensively and therefore succeed in providing an overall high data quality.

Munappy et al. [13] identify quality assessment to be an important part in pre-processing and conducting the validation of data in a separate stage, with possible report creating, machine learning applications and dashboard creating.

Rogojan [106] elaborates on dbt framework [105] as a good solution for providing data quality within pipelines. The framework is easy to use and provides the opportunity to ensure data quality with versioning and database tests (e.g., schema and data integrity tests) in the transform phase of the pipeline [106].

Data Repair

For solving problems with deep neural network fault severity estimation, Yao et al. [82] propose a solution of two-stage data quality improvement strategy in its pipeline. The first stage is developing a spatial information reconstruction approach [82]. In the second stage, spatial signals are organized based on different faults and types of models to aim to find useful information [82]. Although this solution is specific to its domain, Yao et al. [82] signify the benefit of targeted and thoughtfully engineered data on data quality, which also reduces the requirements for model optimization [82]. The results demonstrate a higher data accuracy and robustness under challenging data type and speed conditions [82]. The importance of quality in the initial data over quality in models and transformation process is mentioned also by Pölöskei [114]. A conclusive idea would then be to put less effort on the quality of cleaning and transformation inside the data pipeline and focus more on creating and maintaining high quality data at the beginning of the pipeline flow or even in data sources, if possible.

As [82] used spatial information reconstructions, Catarci and Scannapieco [95] make a similar notion of data repair, so that the data pipeline could modify the assets to reach an acceptable data quality level.

As a form of solution for gap filling, [49] suggests interpolation, correlation with other signals, replacing values with daily averages, repeating values obtained on the preceding day and using models.

Metadata and Data History

Munappy et al. [13] suggest having the original raw data file stored, so it can be accessed whenever processed data does not meet quality requirements.

Therrien et al. [49] mention the importance of keeping thorough metadata from pre-processing and using extensive data versioning with version control to counter situations where issues arise and the progression of the data needs analysis.

Data lineage clarifies the data, its origin, transformations, and attributes [35]. Metadata management can be used to catalogue information about the data such as schema details, data quality metrics, and lineage [35]. Understanding data during processing can provide more efficient ways of finding data quality errors and solving them [35].

Choudhary [88] mentions metadata documentation and data lineage to be crucial in terms of tracking down errors and avoiding data quality issues introduced during specific processes. Automated data discovery and smart tagging can provide efficient help in developing this [88].

Human Involvement

Abdellaoui et al. [50] encourage involving users to interactively validate the clean data before it is loaded into the next stage. The user could even interfere in the process, as in addition to seeing the process, the user could make changes and suggestions [50].

Asghari et al. [111] have implemented extra cleaning steps to enhance data quality. This approach combines metadata with expert knowledge by creating a database to use as an internal lexical in the tokenization process [111]. This approach helps improve data quality by providing a predefined dictionary for the data cleaning stage of the pipeline. Asghari et al. [111] also apply automatic noise detection via a noise cancellation model to improve the data quality.

Hu [55] discusses the importance of including data consumers in the development of data pipelines. When the pipeline developer is not familiar with the metrics they are delivering, the final quality of the data will not be optimal [55]. However, this problem can be reduced by involving domain experts in the development process of the data pipeline [55].

Special Techniques

Pervaiz et al. [25] mention systems that can aid data cleaning, integration, and other calculations with visual interference [102, 103, 115] and a sensemaking model [116]. As an overlapping solution for processing raw data, [25] suggests tools like Wrangler [102] and Google Refine [117] to include users in the cleaning process and use users as a form of conflict resolution to help functions produce higher quality data. Google Refine is also mentioned in [74] as a widely used reconciliation tool to help clean data, along with TIBCO Clarity and Winpure.

Somasundaram [75] employs advanced analytics techniques in the form of machine learning algorithms and statistical analyses in data quality monitoring processes to find anomalies and validate data transformations.

Clayson et al. [76] mention a focus on standardization as a potential optimization technique for ERN (error-related positivity) and Pe (error positivity) in data processing pipelines. The findings from [76] indicate that certain data processing decisions significantly impact optimizing ERP estimates, and the proposed process of examining the data processing multiverse serves as a roadmap for achieving standardization in ERP scores.

Data pipelines need to have data quality management solutions implemented [71]. The solutions can vary from simple data range checks to more advanced checks involving statistical analysis or data frame checks [71].

General Architecture

Husom et al. [71] implemented their data pipeline using Data Version Control (DVC) framework, which supports large files offers robustness designing a complex, data-quality-sensitive pipeline.

Therrien et al. [49] insist on integrating an online automatic fault detection into the pipeline, with also fallback strategies in the form of controller reconfiguration.

Combating the problem of data loss from data transformation, Munappy et al. [18] suggest using a lossless data transformation technique when possible.

c) Load Stage

The two single data loading challenges were related to lack of robustness and domain variance, which both had a single solution.

Robustness

Sopan and Berlin [113] designed a visualization system for monitoring model performance, detecting issues with data, models, or labelling, and enabling the investigation of reasons behind issues. In addition to its main goal of assisting the ML deployment, it helps detect and visualize data quality issues with its the data pipeline [113]. The dashboards have a data quality section that presents data volume and rate, to help data scientists and engineers to have a measure of data quality for the processed data [113].

General Architecture

Ardagna et al. [33] suggest defining alternative deployments with different weights and techniques when catering to a specific need of performance. Data quality can be subjective to its use case and adjusting and different flows can have a more successful effect than defining a jack-of-all-trades-type pipeline. Ardagna et al. [33] also mention summarization as a key concept in providing high-importance information about data quality. Having a summarized measure helps demonstrate the feasibility of data [33].

d) Entire Pipeline

When discussing the design of the entire pipeline, the most occurring topic was to implement some degree of validation or monitoring with optional alarm system to make the pipeline more robust, and its data more observable while also denying decrease in data quality through validation. Other solutions were focused on including metadata and/or data history in decisions, involving humans and testing.

Validation, Monitoring and Alarms

Foidl and Felderer [51] try to solve the problem with unmanageable scope of validation by introducing risk-based testing concept. The risk-based data validation approach focuses on assigning data to risk items based on the likelihood and impact of the risk [51]. Risk items can be prioritized and classified to make functional and operational decisions and make a complex data validation process more feasible [51].

Byabazaire et al. [78] proposed a three-stage big data end-to-end data quality assessment using trust. The first stage focuses on initial trust, meaning the context of the data (i.e. equipment, sensors, metadata). The second stage investigates trust from the data itself, evaluating completeness, accuracy, and the processing the data has already undertaken. The third stage monitors the products using the data and acts as a feedback mechanism to upstream data evaluation. The approach enables data quality estimation without standards and provides a general representation of data quality throughout the pipeline. [78]

Ryza [81] discusses thinking of data quality as a fundamental part of the pipeline, which is not achieved by simply adding data quality checks. Adding orchestration to data quality checks is crucial, and Ryza [81] advertises the flow of Dagster. An interface provides observability to the flow and data quality checks within the pipeline adds the option to debug

and untangle the roots of the problems [81]. Alerts are also an important part of the data pipeline flow, with Dagster also providing high-fidelity information with the alerts and the health of the pipeline [81].

With connectors sending data between data pipeline stages, each one should have fault detection and then the capability of mitigation strategies or sending alarms [13]. Without fault detection operations in connectors, data quality issues could be left unchecked, or there could be data loss between the stages.

Bail [73] advertises dbt framework [105] but makes the general point of alleviating the pressure and hardship of data testing with workflow orchestration tools in general. Airflow[8], dbt and Great Expectations[9], together with Dagster[10], Prefect[11] and Kedro[12] are good options [73]. The frameworks also generate reports to find and investigate data quality issues [73].

One key aspect to maintaining smooth uninterrupted flow of data and quality assurance is monitoring [35]. Tracking and observing pipeline processes will make the detection of anomalies and bottlenecks easier and ensures that the data flaws will not be introduced or missed by the pipeline [35]. Data reliability can also be ensured with automated testing during development, to find flawed techniques that might introduce data quality issues [35]. Similarly, [27] signifies the importance of implementing continuous monitoring, fault detection and mitigation throughout the data pipeline. This can help detect data quality issues before the data reaches the end of the pipeline [27].

Yaddow [107] suggests implementing data quality tools, continuous monitoring, periodic audits, and feedback loops as a form of data quality assurance within a pipeline. Training and educating teams into these processes and staying aligned with regulations and standards can then help enforce the data quality assessment strategy [107]. Documenting, reviewing, and refining the set data quality practices can then further improve the overall quality of data within the pipeline [107].

Data observability holds the key to building data quality assurance within the pipeline. Comprehensive coverage across the pipeline can help detect data quality issues near their roots [44].

Valarezo et al. [80] discuss the benefits of data observability through monitoring and metadata collection. Ensuring data observability allows instant detection of data issues, which decreases delays within pipelines and enables a swift flow for solving the found issues [80]. Observability also provides the pipeline with a continuous approach to data quality checks, minimizing the risk of data quality issues being unfound [80]. Collaboration between teams is important to add observability to add impactful metadata and lineage metrics to the data [80]. Automating data quality tasks, measuring, and observing metrics, and involving business parties in these activities can help solidify a higher level of data quality in the pipeline [80].

Metadata And Data History

Voropaeva [97] suggests data quality assurance within the data pipeline to be heavily dependent on the availability of metadata. The visibility of checks and transformations can

[8] https://airflow.apache.org/
[9] https://greatexpectations.io/
[10] https://dagster.io/
[11] https://www.prefect.io/
[12] https://kedro.org/

not only provide an overview in manual analysis and overview, but also pinpoint the erroneous data or operations more precisely [97].

Sofer [110] named two popular data pipeline data quality assurance practices of versioning with data copies and various testing strategies of unit, integration, and exchange-to-exchange tests. As an enhancement, [110] suggested using a data version control (e.g., lakeFS). This approach increases the data engineering velocity with more efficient data versioning and testing, also providing a good platform for using multiple isolated testing environments [110]. With lakeFS, the isolated test environments would operate on a different server without data copy [110].

Yaddow [107] discusses data governance to play a large part in ensuring data quality within data pipeline projects. Developing data quality assessment should start with establishing the constitution of data quality within the organization through accuracy, completeness, timeliness, consistency, and reliability [107]. Each data quality dimension should have a measurable metric, aligned with expectations and standards [107].

For allowing early detection and reporting of data quality issues, Challen [109] proposes dtracker wrapper package around tidyverse data manipulation tools to allow automatic tracking of processing steps applied to data.

Human Involvement

Data engineering is becoming increasingly popular and data pipelines should acknowledge the high importance of data itself beside the traditional software engineering [52]. The pipeline should contain data-specific roles to satisfy the higher number of tasks related to data governance, data lineage, data observability, and overall data quality [52]. Data management roles such as data reliability engineer and data product manager are relatively new, and it represents the need for new roles within the pipeline in addition to software engineers, who might be inaccurately assigned to many data-related tasks [52].

Biessmann et al. [32] mention using human audits in data pipeline operations for data validation to still be efficient and necessary in many cases. Experts or researchers can easily identify errors from both input and output and are an important asset to use within data validation loop [32].

Voropaeva [97] suggests sending alarms to a target group after an error event in the pipeline. These target groups can be both data pipeline developers and application developers [97].

Somasundaram [75] mentions the implementation of an alerting system in their pipeline to promptly notify responsible roles about issues. This is done in the form of dashboards, emails, and instant messages, depending on the severity of the problem [75].

Testing

As with bad data, Dazzeo [43] offers a solution of finding data pipeline methods to improve by introducing uncertainty into the data. This can help pinpoint problematic functions and processes within the pipeline [43].

Data quality assessment is traditionally performed during data ingestion, but [88] suggests testing data quality throughout the pipeline. Data quality assessment should be continuous and proactive, with measurements going beyond ingestion to different downstream processes [88].

Pulkka [67] modernizes ETL pipeline processes and suggests using the testing capabilities of data build tool (dbt) [105] to write automated testing solutions starting with source data all the way to target tables. Tests include simple null checks but also focus on referential

integrity, accepted values and overlapping validity [67]. The tests can be run when new deployments are made but also daily to help find newly introduced data quality issues regularly [67]. Additionally, [81] mentions dbt to be compatible with Dagster[13] to further enhance the integration of data quality solutions.

Robustness

Munappy et al. [46] talk extensively about the trade-off between robustness and complexity. A robust pipeline is fault-tolerant and quality-securing, yet developers may struggle adding necessary complexities to it with attaining robustness. Munappy et al. [46] empathize with the prioritization of one factor without completely compromising the other.

Ovens [79] signifies adding health checks and monitoring to the pipeline. Adding notifications and issue management solutions for when a part of the pipeline fails, can help data pipeline roles to react to data health problems as they arise [79].

To address message reconciliation challenges, [74] proposes a framework aimed at creating a resilient and reliable data pipeline by implementing monitoring and auditing mechanisms. Within the proposed architecture, components such as auditor, monitor, replicator, and scheduler communicate using messages with error and reconciliation metrics [74]. These metrics have various acceptable limits to enable high data quality transmission over the pipeline [74].

General Architecture

May and Fuller [66] suggest building data pipelines as close to the domain as possible and avoiding outsourcing pipeline development. Knowing the data is key to building pipelines, and there needs to be a significant effort to understand the data and its domain [66]. Without extensive data knowledge, the pipeline cannot ensure data quality properly [66].

Taverna et al. [53] mention the need for regular maintenance and improvements to help the pipeline stay current. Changes outside the pipeline should be monitored, and data pipeline reviews should be scheduled whenever standards or paradigms are changed [53].

Increasing complexity of the pipeline can be managed with utilization of advanced data tracking and visualization tools that help understand pipeline processes and data lineage better [36].

Faragó [108] lists seven methods for effective data quality assessment in ML data pipelines: (i) manual assessment with a labelling or reviewing tool, (ii) comparison to standard, (iii) statistics on datasets (e.g., distribution distance, outlier detection), (iv) statistics via schemas (e.g., Great Expectations [118], TensorFlow Data Validation [92]), (v) model performance of the model that data pipeline was used for, (vi) model confidence (e.g., Cleanlab [119] confidence learning tool), (vii) separate quality prediction models.

4.2.3 Schema Issues

There were no direct solutions offered for schema violations and their absence, but schema drift was tackled by many sources. There were 6 mentions of solutions.

4.2.3.1 Schema Drift

Schema changes are a simple issue by nature and solutions include implementing quality expectations and lineage mapping to the data, or simply looking at data history. Some tools are mentioned to detect schema drift, and mitigation strategies can include adjusting the

[13] https://dagster.io/

pipeline architecture and processes to accommodate for potential changes in schemas. All the solutions were targeted at the extract stage of the pipeline.

There is no universal solution to detecting schema changes, but [44] mentions a combination of data quality checks, lineage mapping, and internal processes to be the key to detecting schema changes and adjusting downstream decisions based on the results.

Yaddow [36] mentions using different tools to track and manage schema changes as a protection against modified data structures introducing errors or reduced data quality within pipelines. Redyuk et al. [70] successfully use TensorFlow [92] to detect data schema violations.

Armbrust and Lappas [85] discuss expectations (i.e. assertions about data) to help identify schema drift and data structure changes within pipelines. Another approach is leveraging data history to detect schema drift [81].

Sadiq et al. [65] lay the groundwork for researching the role of empiricism in data quality assessment. This is reliant on implementing intrinsic metrics (e.g., format-consistency of measurable attributes) and extrinsic metrics (e.g., fidelity of a specific analytical report) [65].

4.2.3.2 Schema Errors

There were no sources directly solving schema errors in data pipelines.

4.2.4 Human Errors

Issues with human involvement in data pipelines were attributed to data entry errors and poor data quality decisions due to lack of expertise. There were 11 mentions of solutions.

4.2.4.1 Data Entry Errors and Lack of Expertise

Data entry errors and lack of expertise are analysed together due to the solutions having the capacity to tackle both issues, and separation of the solutions would not provide a clearer overview.

a) Extract Stage

The solutions and mitigation strategies against human errors in the extract stage are classified into four groups: Standardization, Data Validation, Data Entry Training, and Advanced Techniques.

Standardization

Mitigating human errors in data sources can be achieved through standardization, defining data validation for data creators and their used tools, and establishing data governance to measure the quality of the incoming data [55].

Taverna et al. [53] combat human errors by standardizing the curation process. Checklists, training, and documentation of best practices can improve the implementation of these standards and ensure more consistent data quality [53].

Pervaiz [42] achieves reduced human error through replacing potentially erroneous paper data entry with standardized SMS data reporting in CCIS.

The organizational policy outlined in [62] enhances the qualitative judgment of companies (data sources), ensuring that the extracted data maintains high quality with minimized judgmental errors.

Data Validation

Towards Automated Detection of Data Pipeline Faults [18] is a forerunner for trying to solve the human involvement and human-caused data error problems in designing data pipelines. Munappy et al. [18] introduce several ways to reduce human involvement but in many cases the automations will remain a mitigation strategy and still require some form of human interaction. Data validation is the main mitigation strategy to find human errors as it can perform checks to detect errors as soon as intended [18]. Automated detection of faults and data validation will help reduce human-error data throughout the data pipeline and therefore potentially the quality of data within the pipeline.

Aaron [19] suggests tackling human errors in their root. For instance, [19] suggests implementing data validation within data sources to prevent employees from consistently using different formats.

Osborn [44] mentions RegEx as a useful tool for validating common patterns with phone numbers, emails, numbers, escape characters and dates.

Data Entry Training

Aaron [19] mentions training users to improve their data literacy. When users understand data, data tools, formats, and processes, they can attribute to a more careful and precise approach in data entry [19].

Advanced Techniques

In response to the challenges posed by the lack of resources and inadequate infrastructure in developing regions, where large-scale data collection relies on paper forms, researchers have devised innovative solutions [25]. The solutions include utilizing smartphone cameras for digital capture and leveraging cloud and crowd-based approaches to process forms more efficiently [25].

b) Transform Stage

There are no sources targeting the transform stage in solving challenges related to data entry errors and lack of expertise.

c) Load Stage

There are no sources targeting the load stage in solving challenges related to data entry errors and lack of expertise.

d) Entire Pipeline

The solutions and mitigation strategies against human errors in the entire pipeline are classified into two groups: Data Quality Training and Visualization and Automation.

Data Quality Training

For solving an issue with developing a precise data pipeline with data quality in mind and reducing the gap between data scientists and domain professionals, [49] suggests strong collaboration, extensive use of domain-specific knowledge and training.

Visualization and Automation

To encourage humans to produce more accurate data and conclusions, [49] recommends using adaptive interfaces, interactive dashboards, visualizations, and reports, along with colour, shape, and spatial placement. Computers think with math, but humans do not [49].

Moreover, [85] signifies the importance of automation and reduction of manual tasks as much as possible to reduce erroneous human involvement in pipelines.

Somasundaram [75] mentions an alerting system with dashboards and real-time visualization of metrics in their monitoring process to provide effective data observability.

4.2.5 Data Loss

Solutions for identifying events where data was delayed or drifted from its intended path were similar in their nature, and the strategies to pinpoint and repair this behaviour are therefore handled together. There were 10 mentions of solutions.

4.2.5.1 Data Loss, Data Downtime, Hidden Data and Data Leaks

The key to solving issues regarding data loss, data downtime, hidden data, and data leaks, is promptly detecting the moment of occurrence, finding the point of origin, and eventually reacting to the problem. This requires implementing strategies such as continuous monitoring, integrating fault-detection mechanisms, and activating alarm systems to alert stakeholders of potential issues. The solutions have a distinct nature and are not further categorized within each stage.

a) Extract Stage

Munappy et al. [18] try to mitigate data loss inducing problems of data source failure, inactive data source, authentication failure or data sending job failure by sending an alarm when a device or source fails or notification for reactivation when possible. However, this introduces human intervention, and it remains only a mitigation strategy.

Monitoring mechanisms can help mitigate issues with data downtime, as it enables interference as soon as data is not being received [19]. Munappy et al. [13] also mention tracking mechanisms for failures to help identify data loss.

In addressing data loss arising from malfunctioning sensors or non-compliant devices, [39] proposes a solution that involves initially assessing the sensors themselves when data loss is detected at any point.

Roman et al. [23] mention interpolation as a solution to when a data is missing for a very short period. However, for prolonged gaps in the data, it is advisable to not even include the data into the analysis, particularly with machine learning analytics [23].

b) Transform Stage

There are no sources targeting the transform stage in solving challenges related to data loss, data downtime, hidden data, or data leaks.

c) Load Stage

There are no sources targeting the load stage in solving challenges related to data loss, data downtime, hidden data, or data leaks.

d) Entire Pipeline

Regarding data loss from node-to-node transmission within all data pipeline stages, [5] suggests adding a fault detection mechanism to identify the origin of disappearance. Logging or saving the point of disappearance can help mitigate the issue and potentially avoid complete data loss. Designing a highly traceable data pipeline is heavily

recommended not just for the origin of data loss or error but also as a general good practice [5].

Suleykin and Panfilov [120] point out the importance of constantly checking the status of all ETL pipeline layers to detect data loss. Constantly validating that the correct amount of data records has moved between layers is crucial in finding erroneous stages and lost data. For the big data pipeline described in [120], there is a Technical Data Quality Agent responsible for checking data completeness for each storage layer.

Munappy et al. [46] signify the need for continuous monitoring and fault detection to prevent data leakage and maintain data quality. Somasundaram [75] incorporates fault detection mechanisms, automated recovery processes and backup strategies to minimize data loss and downtime. This allows data loss detection and enhances data quality in providing data completeness.

Hilleary [89] suggests using machine learning to analyse data flows and to generate rules for identifying data leaks within the pipeline. DataBuck[14] is one example of an automated data quality platform that uses machine learning to identify issues regarding data leakage [89].

4.2.6 Scalability

Both technical and operational scalability issues have similar solutions, as both strive towards efficient automation. There were 15 mentions of solutions.

4.2.6.1 Technical and Operational Scalability

The solutions have similar intent, but technical scalability goes beyond automation by also focusing on the efficiency of the used techniques and environments. The solutions include both popular and custom automation frameworks, cloud environments, and using very specific techniques or algorithms for processing data. The solutions are mostly general, but some specific solutions are exclusive to solving challenges with technical scalability.

a) Extract Stage

Like with assuring data quality with raw data, Sakdas [47] can also assist in maintaining data quality with increased data pipeline capacity. Sakdas can solve scalability issues regarding developing extensive data pipelines with limited development resources where data quality assurance cannot be solved with large and industry-specific data quality frameworks.

Ardagna et al. [33] signify the importance of automation, and including adaptation techniques together with predefined patterns within the pipeline for when there is an increase in heterogeneous data sources. Yaddow [36] suggests considering the potential differences between data sources when implementing data quality monitoring tools. The search for inconsistencies, duplicates or missing values should be applicable to any data source [36]. Redundancy and failover mechanisms with continuous health checks can help the pipeline to run smoothly as well [36].

Expanding on [95] solving raw data extraction issues, Catarci and Scannapieco also signify OBDA's automated inference capabilities to detect data inconsistencies, providing a system-wide standard and optimized algorithms for scalability with large datasets.

[14] https://firsteigen.com/

b) Transform Stage

DataWig [37] is a scalable approach to data imputation with hyperparameter tuning that requires minimal effort in tables with heterogeneous data types, reducing the effort towards designing specific data imputation method for each data type.

With retaining high data-quality in mind, Pölöskei [114] suggests cloud-based solutions to provide data processing efficiency compared to an on-premises environment. Pölöskei [114] uses predictive analytics for supporting data quality assurance and cloud-based solutions can be an option to scale data pipelines as the quality assurance within the transformation becomes increasingly complex.

Huang et al. [41] implement on-line cluster sampling algorithm to ensure that the data points are evenly sampled across the full spectrum of the distribution, regardless of the population of specific workload patterns. The method aims to maintain efficiency and diversity in labelling even as the complexity and variety of data increases, allowing retained high data quality [41].

c) Load Stage

There are no sources targeting the load stage in solving challenges related to technical or operational scalability.

d) Entire Pipeline

Ryza [81] mentions using asset check factories with Dagster or dbt to reduce the effort going towards building data quality checks. The checks are usually similar in their core, so having an opportunity to use a factory can make the entire process more manageable for all roles within the pipeline [81]. Similarity, [34] mentions data orchestration tools of dbt, Airflow and Great Expectations to provide a platform for automated profiling. This can help define data validation based on historical data and reduce effort in building or scaling data pipelines.

Song and He [84] have developed Auto-Validate, an unsupervised data validation framework which works on single-column constraints using patterns. Auto-Validate determines whether a pattern is suitable for the column from pre-existing patterns inferred from data lakes [84]. Auto-Validate framework makes data validation in data pipelines scalable when the need for writing data constraints surpasses the possibility and ability to write them manually.

Cold Chain Information System [42], targeted at low-resource environments, provides health facilities with an opportunity to make and view reports to help data pipelines scale to the national level.

Byabazaire et al. [90] propose integrating fusion methods into end-to-end data quality assessment to serve different applications within a single pipeline, as opposed to each application having a separate data pipeline. The process initiates with the first core component, the real-time Data Quality Assessment (DQA), which operates through three stages. Each stage focused on unique data quality dimensions such as accuracy, completeness, consistency, and timeliness. The DQA assesses data quality in real-time, with results from each stage feeding into the second core component, the Data Fusion Engine. This component consolidates intermediate quality assessments, employing different fusion strategies to yield a single quality score. Both components are modular, allowing independent application of fusion strategies and ensuring a comprehensive evaluation of data quality across various dimensions to support diverse application needs. [90]

Wallace [35] mentions cloud environments to provide scalability in terms of serverless computing models, dynamic resource scaling and data storage optimization. This addresses the problem with retaining data quality assurances in the form metadata management, monitoring, and functions of data classification, analytics, or anomaly detection [35]. Containerization and orchestration can also enhance scalability and resource utilization in the same way [35]. Similarly, [53] mentions cloud-optimization to improve computational performance and storage cost efficiency, which contributes towards improved scalability.

Merino et al. [91] suggest building data quality assessment independently and separate from the main pipeline to have the least disruptive approach to data quality evaluation upon pipeline expansion.

4.2.7 Data Drift

Data drift refers to the gradual deviation of data characteristics over time, and the solutions discussed in this section tried to capture techniques for identifying the deviation, as normal validation does not detect this. There were 12 mentions of solutions.

4.2.7.1 Data Drift

Recognizing data drifts and anomalies is challenging and was found to require vastly different technical solutions. There were different tools mentioned that specialize in drift and anomaly detection, yet for more specific situations, sources suggested alerts for suspicious data, continuous monitoring, and machine learning. Another key was to detect these issues early before they have cascading effects within the pipeline and further.

a) Extract Stage

Yaddow [36] mentions implementing domain-specific anomaly detection and isolation mechanisms to mitigate erroneous data flows in streaming data. Machine learning-based algorithms can identify and alert new data patterns and data quality issues [36].

Redyuk et al. [70] describe an approach of training a model to detect novelties in data and using it to produce a data quality measure to accept kind of data into the pipeline. Data pipelines can raise alerts about suspicious data that shows degradation signs, helping developers and domain experts to intervene [70].

Therrien et al. [49] emphasize the importance of devising a strategy for maintaining sensors and the data they produce over their lifespan, aiming to identify and address any inconsistencies or losses in the data. These inconsistencies can also be found with anomaly detection methods of both open source and commercial availability, depending on the domain and use case [32].

In discussing data drift, Song and He [84] suggest taking after large tech companies by catching the issue as early as possible in the pipeline. This approach relies on using data validation tools [84]. Google's TensorFlow Data Validation [92] and Amazon's Deequ [93] are a couple of known tools for developers to write domain specific data constraints to describe "normal" data, which can in turn assist in catching unexpected deviation as it is being introduced into the data [84]. The tools offer very high-level specifications of constraints, resulting in highly scalable data validation [84].

Ryza [81] elaborates on the creation and orchestration of drift detection. In Dagster, it is possible to define anomaly detection based on metadata [81]. Previously gathered metadata from asset materializations on each run of the pipeline can help define a data quality check

that calculates mean and standard deviation and compares the numbers with the incoming data to recognize data drift [81].

Redyuk et al. [70] highlight Deequ [93], TensorFlow [92], and general statistical testing to be efficient in detecting data drift. Redyuk et al. [70] also introduce their own data quality validation approach to help detect drift during data ingestion.

Osborn [44] suggests writing distribution tests, by providing minimum and maximum values to columns. A couple of examples include *accepted_values* test of dbt [105] or unit tests of Great Expectations [44].

b) Transform Stage

Mattila [38] mentions the importance of continuously monitoring data pipelines to recognize data drifts and changes in data. Acceptable values should be defined in rules, and different metrics in the form of mean and standard deviation should be pre-computed [38]. The complexity of the rules, metrics and techniques should be in accordance with the complexity of the data and its purpose.

Tu et al. [83] developed Auto-Validate-by-History framework to automate validation in recurring data pipelines. The framework can recognize data quality deviations, allowing data pipelines to easily detect data drifts. Therrien et al. [49] also suggest data reconciliation frameworks and using signal features that are trusted even with unmaintained or faulty signals.

Huang et al. [41] developed drift detection for tackling data distribution drift and its decreasing effect on data quality over time. This allows the workload clusters to be recomputed and removes the need for more complex algorithms [41].

c) Load Stage

There are no sources targeting the load stage in solving challenges related to data drift.

d) Entire Pipeline

There are no sources targeting the entire pipeline in solving challenges related to data drift.

4.2.8 Volume

Data volume related solutions are divided into two sections of solving high and low volume issues that cause data quality problems within data pipelines. Data high volume is the more popular problem and has considerably more coverage compared to low volume.

4.2.8.1 High Volume

To mitigate the impact of high volumes of data on both the pipeline and data quality, solutions such as volume tests, load balancing strategies, and data partitioning and sharding techniques are employed. Another common approach was to conduct exhaustive data quality checks after the transformation to avoid disrupting the main flow.

a) Extract Stage

Osborn [44] suggests implementing volume tests. These tests would identify data volume changes, and therefore help uncover compromised data models or decrease the volume to a specific needed point [44].

b) Transform Stage

There are no sources targeting the transform stage in solving challenges related to high volume.

c) Load Stage

There are no sources targeting the load stage in solving challenges related to high volume.

d) Entire Pipeline

To avoid data quality issues derived from decreased performance, data quality evaluation should not introduce latency issues in the main flow of the pipeline [30]. In data stream applications, it is common to check data quality after the transformation and load has been finished [30]. This allows implementing risk strategies and decisions after the data monitoring is complete, in parallel with the main flow while not causing interruptions [30].

Yaddow [36] mentions using scalable data quality monitoring architectures and load balancing solutions to handle high data volumes. Data partitioning and sharding techniques can also help reduce issues with high volume in the form of data distribution and parallel processing [35].

4.2.8.2 Low Volume

For problems regarding small datasets during data extraction, [23] suggests sharing the data, as community-sourced datasets help in increasing the value of the data. This could however introduce a whole different entity of problems regarding data quality in the extraction or transformation stages of the pipeline.

4.2.9 Compatibility

Solutions regarding system and hardware incompatibilities are varied due to the distinct nature of each compatibility problem. There are still enough references through 6 sources to provide a foundation of solutions and mitigation strategies.

4.2.9.1 System and Hardware Incompatibilities

While the nature of incompatibilities surrounding and within the data pipeline processes can vary, some solutions include using standardized protocols and formats, tracking updates and changes, and using validation. Additionally, semi-structured models can provide looser and more robust connections between systems.

a) Extract Stage

In the past, the solution for solving incompatibilities between devices and applications in data collection was typically done with custom drivers and proprietary communication protocols [49]. However, more recently the prevailing tendency leans towards achieving interoperability through standardized and openly accessible digital communication protocols [49].

Taverna et al. [53] mention defining flexible solutions in data integrations, to allow robustness while adding data sources with incompatible standard formats. This means designing integration for the most common formats so slight variations would not break the pipeline [53].

b) Transform Stage

Regarding performance data quality issues when there are updates or changes in hardware or software, [41] suggested distinguishing performance drifts and tracking of monthly system performance. This allows engineers to take proper action when updates or changes result in data quality dip through performance data drifts.

c) Load Stage

Munappy et al. [18] suggest updating the data parser or even defining a new one as a mitigation strategy against data sent from incompatible devices. To detect issues with incompatibility, [18] suggests adding data validation and using data imputation techniques.

d) Entire Pipeline

Sacolick [60] suggests using data stores such as data lakes and semi-structured data models in graph databases, as they help reduce data debt and connect data in a looser way, thus reducing compatibility and integration problems.

For storing big data, [49] recommends looking at data warehouses and data lakes instead of traditional simplistic databases to solve possible data storage incompatibilities that can arise in the data pipeline.

4.2.10 Logical Errors

Solutions regarding logical errors were often architectural improvements to avoid the issues completely due to their random nature. There were 6 mentions of solutions.

4.2.10.1 Logical Errors

Logical errors are very straight-forward in their root causes, yet sometimes they difficult to find. Since the issues can vary greatly, there is an importance on developing the pipeline with a clear intent, and continuous monitoring to catch potential discrepancies that are introduced due to changes outside the pipeline. Time synchronization, access issues, and the introduction of new data sources need to be well thought out for them to not cause cascading effects.

a) Extract Stage

In solving logical errors that are caused by introducing a new entity to the data pipeline, Munappy et al. [18] suggest sending out a notification. For example, when a new data source is introduced, there needs to be a notification to the concerned data pipeline owner to take action to accommodate the new data by building a new ingestion module for the new data source [18].

With solving minor logical inconsistencies with timestamps, [23] uses average and linear interpolation fix the data quality issue and points out the possibility for the usage of more advanced methods. For stream processing timestamp issues, [23] buffers the data for extended periods to wait for every sensor to upload their data. It is a solution that enhances data quality, but at the expense of time.

Therrien et al. [49] underscore the importance of considering the entire data pipeline ecosystem and its thoughtful design. The data pipeline needs to be developed with a clear intent to minimize future data quality issues arising from logical errors in its flow. Ambiguously designed data pipeline can result in data graveyards [49].

b) Transform Stage

Logical errors are mostly caused by changes outside the data pipeline. For achieving constant data quality in transformation stage, the data pipeline must continuously be monitored for data drifts and change in data distribution [5]. It is also important to update the data pipeline and its business logic as frequently as there are changes in data sources [5].

Since misinterpreted data is closely related to unclear definitions, a mitigation strategy for correct interpretations and clear definitions can simply be to contact subject matter experts [18]. Understanding the specific data aspects and transformation process more deeply can be achieved by clarifying with experts and this can help achieve higher quality of data.

c) Load Stage

There are no sources targeting the load stage in solving challenges related to logical errors.

d) Entire Pipeline

Therrien et al. [49] signify the importance of increased collaboration between parties to solve problems that impede access to data.

4.2.11 Security

The main data quality challenges stemming from security issues were related to authentication. The key to mitigating this issue is designing proper authorization throughout the pipeline and its starting and ending points.

4.2.11.1 Unauthorized Access

Security solutions within data pipelines usually require specific implementation. The problem is generally solved by implementing security measures and does not have too much depth to it. The few data-quality focused solutions to security issues within data pipelines involved designing authentication for not only the pipeline processes, but also the data itself through history. The following solutions were not specifically focusing on any ETL stage.

Munappy et al. [46] mention the implementation of authentication processes throughout the pipeline to validate the rights of access and protect against identity theft and fraud. From a data quality perspective, it ensures that data is modified only by authorized persons and protects against external attacks [46].

Byabazaire et al. [57] mention blockchain approach together with data quality assessment models to provide a measure of trust. As data can move and be affected vigorously, this approach adds *trustability* and history to data or datasets [57].

4.3 Open Issues

The open issues are determined by comparing the mentions for issues and solutions. In addition, some sources explicitly elaborated on future work, providing extra insight into challenges that need to be addressed. Table 25 contains an overview of all the mentioned challenges, with Table 26 concluding the corresponding information on solutions and mitigation strategies. The following sub-sections conclude each challenge and the outlook on future work required.

Table 25 Challenges Overview

Challenge	Sub-challenge	Stage	Number of mentions	Total	
Bad Data	Raw Data	Extract	22	35	70
		Transform	8		
		Entire Pipeline	5		
	Data Trustworthiness	Extract	22	35	
		Transform	6		
		Load	2		
		Entire Pipeline	5		
Pipeline Architecture	Lack of Robustness	Transform	3	10	37
		Load	1		
		Entire Pipeline	6		
	Lack of Domain Knowledge and Expertise	Transform	3	10	
		Load	1		
		Entire Pipeline	6		
	Lack of Observability	Transform	3	10	
		Entire Pipeline	7		
	Lack of Validation	Transform	5	7	
		Entire Pipeline	2		
Schema Issues	Schema Drift	Extract	10	13	17
		Load	1		
		Entire Pipeline	2		
	Schema Errors	Transform	2	4	
		Entire Pipeline	2		
Human Errors	Data Entry Errors	Extract	9	9	17
	Lack of Expertise	Extract	1	8	
		Transform	2		
		Entire Pipeline	5		
Data Loss	Data Loss	Extract	4	9	16
		Transform	3		
		Entire Pipeline	2		
	Data Downtime	Extract	1	5	
		Transform	1		
		Entire Pipeline	3		
	Hidden Data and Data Leaks	Extract	1	2	
		Entire Pipeline	1		
Scalability	Technical Scalability	Extract	5	11	15
		Transform	1		
		Entire Pipeline	5		
	Operational Scalability	Extract	1	4	
		Entire Pipeline	3		
Data Drift	Data Drift	Extract	7	14	14
		Transform	6		
		Entire Pipeline	1		
Volume	High Volume	Extract	4	9	11
		Transform	3		
		Entire Pipeline	2		
	Low Volume	Extract	2	2	
Compatibility	System and Hardware Incompatibilities	Extract	2	9	9
		Transform	2		
		Load	1		
		Entire Pipeline	4		
Logical Errors	Logical Errors	Extract	4	7	7
		Transform	2		
		Entire Pipeline	1		
Security	Unauthorized Access	Entire Pipeline	6	6	6

Table 26 Solutions Overview

Challenge	Sub-challenge	Stage	Number of mentions	Total
Bad Data	Raw Data Data Trustworthiness	Extract	33	67
		Transform	17	
		Entire Pipeline	17	
Pipeline Architecture	Lack of Robustness Lack of Domain Knowledge and Expertise Lack of Observability Lack of Validation	Transform	23	55
		Load	2	
		Entire Pipeline	30	
Data Drift	Data Drift	Extract	4	12
		Transform	4	
		Entire Pipeline	4	
Schema Issues	Schema Drift	Extract	6	6
		Load	0	
		Entire Pipeline	0	
	Schema Errors	Transform	0	
		Entire Pipeline	0	
Data Loss	Data Loss Data Downtime Hidden Data and Data Leaks	Extract	5	10
		Transform	0	
		Entire Pipeline	5	
Volume	High Volume	Extract	1	5
		Transform	0	
		Entire Pipeline	3	
	Low Volume	Extract	1	
Scalability	Technical Scalability Operational Scalability	Extract	4	15
		Transform	3	
		Entire Pipeline	8	
Logical Errors	Logical Errors	Extract	3	6
		Transform	2	
		Entire Pipeline	1	
Compatibility	System and Hardware Incompatibilities	Extract	2	6
		Transform	1	
		Load	1	
		Entire Pipeline	2	
Human Errors	Data Entry Errors Lack of Expertise	Extract	9	11
		Transform	0	
		Entire Pipeline	2	
Security	Unauthorized Access	Entire Pipeline	2	2

4.3.1 Bad Data

Bad data is the most mentioned problem in the analysed sources with 70 total mentions across all the stages (32%). It has a lot of coverage in terms of solvency, through 67 mentions. Most of the data quality issues are related to data ingestion and extraction phase, because of the raw nature of the incoming data and its untrustworthy nature. While having a high solvency in terms of solutions-per-mention, the extraction stage is the one with lowest solvency of 75%. Since bad data is most mentioned, and most prevalent during data extraction, it provides direction for future research. In addition to the data ingestion phase, a large focus should be on establishing the highest data quality already within the initial data sources that create and collect the data.

Another unsolved part of the bad data was the late-stage challenges with the data lacking quality measures when loaded into the target system. The need for data quality measures throughout the pipeline is also inherent to data trustworthiness issues, which is something that needs further research in terms of solutions.

The issues with bad data are heavily related to [2], which takes a specific focus on data issues within data pipelines. Foidl et al. [2] have found comparable results of data type faults and raw data, but also deeper issues with symbols, characters, and data frames.

The most efficient approach in solving the plentiful different data issues within data pipelines might be to investigate data-focused strategies and have less focus on the pipeline itself. This thesis was limited to staying within the bounds of data pipelines, yet the deep-root causes of missing values, outliers and data type issues could be tackled by implementing general well-known data practices and techniques.

Approaches that include scalable heuristics and ML based data type inference techniques could provide efficient solutions to restrictive data types and encourage data engineers to not use generic data types [32]. Additionally, finding an acceptable combination of algorithms for data deduplication remains another complex combinatorial problem, and an unresolved challenge in some data pipelines [40].

Byabazaire et al. [57] have established the need for future work in terms of frameworks to assess data quality throughout shared IoT systems. Calculating data quality score, together with enabling its storing and processing can induce advancements in ensuring data quality. Moreover, developing mechanisms for quality standard feedback on used data and processes, and subsequently applying them to data in the form of learning is another open challenge. Finally, an assessment framework that would not require new data quality dimensions upon changes in data would allow subjective handling and allow data quality representation in a general manner in big data models and its data pipeline.

Golendukhina et al. [22] propose future work in developing techniques to detect data smells. The investigation of relationships between raw data and the number of data smells they produce is important in improving data quality within data pipelines [22].

Sadiq et al. [65] have outlined the role of empiricism in data quality, and their proposed dimensions can guide data quality research by exposing limitations, gaps, and opportunities. Continued work with more significant focus on data pipelines could assist in defining innovative approaches of ensuring data quality.

4.3.2 Pipeline Architecture

While architectural issues were mentioned 37 times (17%), they are not considered unsolvable because of their straight-forward nature. Designing pipelines in a robust way with validation and monitoring, while involving domain experts is inherently a self-solving challenge. This is further complimented by the high solutions-per-mention rate of 149% and does not require further excessive research. Munappy et al. [13] have also published a comprehensive guide that focuses on the intricacies of data pipeline modelling. However, performance monitoring, data profile monitoring, and condition monitoring remain as open issues in improving traceability of data problems within data pipelines [13].

Golendukhina et al. [22] determine the need for identifying pipeline patterns that create data smells, with the specific focus on data transformation and enrichment phases, where most data smells are introduced.

4.3.3 Schema Issues

Schema issues were mentioned 17 times (8%). There is a low solvency with schema-related issues of 35% with only 6 mentions. Schema drift is the one issue that needs more coverage and attention within data pipelines. Other general schema issues within data pipeline context that need additional attention include schema reverse engineering, handling schema violations and building pipelines in a schema-less way.

4.3.4 Human Errors

Data quality issues caused by lack of expertise or faulty data entry were mentioned 17 times (8%). The 11 solutions offered a solution-per-mention rate of 65%, which is inherently good enough, as the issue is not as complex in its nature. However, human errors are the deep-root reason where bad data issues arise, and they can and should be handled closely.

4.3.5 Data Loss

Data loss, downtime, hidden data, and data leaks were mentioned 16 times (7%). The issues received decent coverage with 10 mentions of solutions, but the issue is one of the more unsolvable ones, as Barr Moses [52] discusses its complicated nature with numerous companies not having concrete solutions for data downtime.

4.3.6 Scalability

Data pipeline scaling affecting data quality was mentioned 15 times (7%). However, the issue had the exact same number of solutions, covering the issue completely.

4.3.7 Data Drift

Data drift was mentioned 14 times (6%), and the problem is very distinct in its nature. The issue has a high solvency rate of 86% with 12 mentions. Further research is not needed due to the small number of mentions and the comparable number of proposed solutions.

4.3.8 Volume

Volume-related issues were mentioned 11 times (5%), with a high volume of data being mentioned 9 times and low volume of data being mentioned twice. The solvency rates are not good as only 4 sources elaborated on solutions for high data volume, and a lone source discussed low volume mitigation strategies.

Data volume issues within pipelines should be further researched, as the loss in data quality comes from processes simply not being to handle the high volume of data or quality-assurance methods not being efficient enough. Processes failing at these tasks often cause data quality loss through data loss or downtime, and this problem can be tackled together with other data loss issues as well.

4.3.9 Compatibility

Data quality issues caused by lack of compatibility were mentioned 9 times (4%). There were 6 mentions of solutions. The issues are quite unique, and they are often boiled down to choosing the correct systems or solving the incompatibilities from updates. Byabazaire et al. [90] establish a basis for future work in terms of scalability considerations, and lack of detailed requirements in practical IoT settings.

4.3.10 Logical Errors

Logical errors are very straight-forward in their nature, and they were mentioned 7 times (3%). The issues had 6 mentions of solutions, further reducing their need for further research. These are simply problems that have caused data quality issues due to lack of planning and thought, and to solve them, they mostly need extremely specific and comprehendible solutions.

4.3.11 Security

Security-related data quality issues were mentioned 6 times (3%). These issues only had 2 mentions of solutions. However, the low solvency is not a deep issue, as the problems were purely related to lack of authentication or security measures. Few sources were aware of solving security issues from a data quality perspective, and there are many potential solutions for improving security within pipelines or between systems.

5 Discussion

This section discusses research findings, limitations, and future work.

5.1 Research Findings

The challenges found throughout the multivocal literature review successfully mapped the most current scenery of data quality issues within data pipelines. RQ1 focused on finding challenges in ensuring high data quality in data pipelines, and RQ2 reacted to the challenges with solutions and mitigation strategies. The challenges were categorized into eleven larger groups. Quite obviously, bad data itself is the core of data quality issues (32%), yet many other unexpected factors also cause difficulties in maintaining constant data quality.

The raw nature of the data entering the pipeline can introduce immediate data quality issues, and it can also cause issues in the later stages. Separately, the lack of data quality measures adds a dimension of untrustworthiness to the data, making it challenging for the pipeline to make data quality-oriented decisions. The specific challenges within these two categories have similarities to the findings of Foidl et al. [2]. The thesis found many solutions to both problems through data quality evaluation techniques and validation, standardization, thorough pipeline design, ontology, data provenance, technical and operational methods, external validation, data processing improvements, monitoring, testing, alert systems, machine learning, and data-oriented roles.

The second-largest group of data quality issues involved the architecture of the data pipeline (17%), with lack of robustness, expertise, observability, and validation. These problems can often stem indirectly from inaccurate data or errors, yet their fundamental source typically lies within the flawed design of the data pipeline. The solutions and mitigation strategies were related to validation, monitoring and alarms, metadata and data history, additional human involvement, testing, improving pipeline robustness, data repair, and various general architectural improvements or use of specific processing techniques. These solutions have a unique nature to them due to their capability of solving other challenges as well. Crafting the pipeline in a specific manner can also help avoid or mitigate numerous separate issues.

Schema issues (8%) were problematic due to their low solutions-per-mention ratio. Schema drift and schema errors received a considerable amount of attention in the sources, yet not enough strategies to combat the issues. This renders it a promising area for future exploration into methods for identifying schema consistency within data pipelines and enhancing data quality in the process. Human errors (8%) were as relevant as schema issues, but they were solved more efficiently through standardization, data validation, employee training, visualization, and automation.

Data loss, together with data downtime, hidden data, and data leaks (7%) caused data quality issues mostly through data incompleteness. These issues are considered critical due to their least controllable nature. The mentioned mitigation strategies included continuous monitoring, fault-detection mechanisms, and alert systems. Surprisingly, incompatibilities between systems (7%) were also mentioned to cause data quality issues, mainly through data loss. While data loss was critical, data drift (6%) was also a considerable challenge due to its unidentifiable nature, as the issue is not detected by the pipeline without employing special techniques and tools, continuous monitoring, or alerts.

Another chunk of problems was caused by high or low data volumes (5%) and scalability (4%) needs. High amount of data flowing through the pipeline is a subproblem of scaling, yet these issues were still distinct enough in their nature and should be tackled independently. Employing more efficient techniques and algorithms often solved both

issues, while automation strategies and introducing cloud environments were also mentioned to improve the situation.

Less significant challenges influencing data quality within data pipelines were general logical errors (3%) and security issues (3%). Logical errors introduced data quality loss through sheer oversights in pipeline design and decision-making, extending beyond mere architectural misjudgements. These errors are however straight-forward in their nature with comprehendible distinct solutions. The same goes for security threats related to unauthorized access, which are often solved through implementing proper authentication and ensuring that the data moving through the pipeline does not lose its integrity.

5.2 Limitations

The thesis employs a multivocal literature review approach, incorporating grey literature into the research. Despite this inclusion, the integrity of the findings remains robust, as the author verified every source, mitigating any potential threat to validity. The main weakness comes from the increasing deviation from data quality and data pipeline contexts as more sources were included. This was mainly the problem with academic sources, as grey literature was kept to a minimum. At times, it was not exactly clear if the found challenges caused issues within data pipelines, and if they compromised data quality in any capacity. Inclusion and quality criteria were crucial in retaining accuracy, yet some bias remains as no significant cross-checking was done.

Another shortcoming stems from the classification of the found challenges. The author found it important to keep challenges in one piece as much as possible, as dissecting and presenting issues under different categories could have produced findings that were unclear or difficult to interpret. For example, an issue with the nature of raw data, caused by missing values and incorrect data types could be classified as two separate challenges. However, all sources were not as specific to conduct deeper classification. Additionally, splitting some challenges could have taken away clarity from the results, while also introducing slight duplication.

Researcher bias through subjectivity can be especially present in RQ1 and RQ2, where the data quality challenges and solutions were extracted from papers that did not exclusively focus on challenges and solutions. Additionally, the papers did not always exclusively focus on the pipeline stage, yet the stage could be determined through logic. To not delve too deeply into finding stages for each issue, the author introduced a fourth category that oversees the entirety of the pipeline.

The cutoff points chosen in 3.1.2 certainly excluded relevant sources. This is especially relevant for Google Scholar results, where the cutoff point was chosen due to the decreasing inclusion rate and increasing effort in determining the inclusion. The cutoff points for grey literature have less impact because of the intent to limit grey literature. However, not all relevant grey literature was included due to employing cutoff points based on titles and descriptions. The author finds that the current proportions in the chosen white and grey literature are highly optimal, yet the work could be more extensive.

5.3 Future Work

Potential future work is mostly discussed in 4.3 section, which tackles RQ3 of finding open issues based on the found challenges and solutions, but also directly from the sources. Bad data is the most mentioned issue that can be researched in greater detail, and it has also been analysed by Foidl in another multivocal literature review [2]. However, solutions to these direct data issues can and should be studied and organized more deeply. Bad data is often

created due to human errors or lack of expertise as well, which was another issue that did not receive enough coverage in terms of solutions and mitigation strategies.

Schema-related challenges represent promising avenues for future investigation, as there exists minimal exploration of schema strategies tailored specifically for data pipelines. Same goes for data volume issues, as both high and low volume caused data quality issues of quite challenging nature, and direct solutions were not often covered. Data loss and data downtime were often left needing more attention as well, due to their indirect yet significant impact on data quality.

Moreover, this thesis provides a foundation for implementing data pipeline frameworks and standardization focused on ensuring data quality. Current work on data pipeline implementation and guidelines is limited, and data quality is often overlooked.

6 Conclusion

The goal of this thesis was to find challenges that are affecting data quality within data pipelines. The multivocal literature review concludes with 89 selected sources, and 219 mentions of data quality challenges.

The first research question focused on finding various data quality issues from sources and classified them into eleven groups to provide an informative aggregate view of data quality issues in data pipelines. The data quality challenges were related to bad data, bad pipeline architecture, schema issues, human errors, data loss, scalability issues, data drift, data volume issues, incompatibilities, logical errors, and security issues.

The second research question investigated solutions and mitigation strategies for the identified data quality challenges. More popular strategies involved implementing validation, monitoring, standards, alerts, data quality assessment frameworks, automation, testing, and many specific algorithms. Many strategies proved effective in addressing multiple challenges, especially those improving the entire pipeline architecture.

The third and final research question found the most unattended issues based on the previous research questions, and directly from the sources. The data quality issues that were left most conspicuous stem from the raw or untrustworthy nature of the data itself, but they were also related to schemas, data volume, and data loss.

The thesis serves as a milestone in outlining the status of key data quality issues within pipelines and offers strategies for addressing these challenges while also identifying areas that may require specific attention.

7 References

[1] "AWS Amazon Announcement," [Online]. Available: https://aws.amazon.com/about-aws/whats-new/2012/12/20/announcing-aws-data-pipeline/. [Accessed 15/4/2024].

[2] H. Foidl, V. Golendukhina, M. Felderer and R. Ramler, "Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers," *Journal of Systems and Software*, vol. 207, no. C, 1 2024. Available: https://doi.org/10.1016/j.jss.2023.111855.

[3] "Batch Processing vs Real Time Data Streams," Confluent, [Online]. Available: https://www.confluent.io/learn/batch-vs-real-time-data-processing/. [Accessed 24/4/2024].

[4] "Snowflake Data Pipeline," [Online]. Available: https://www.snowflake.com/guides/data-pipeline. [Accessed 24/4/2024].

[5] A. R. Munappy, J. Bosch and H. H. Olsson, "Data Pipeline Management in Practice: Challenges and Opportunities," in *International Conference on Product-Focused Software Process Improvement*, Turin, 2020. Available: https://doi.org/10.1007/978-3-030-64148-1_11.

[6] "IBM Data Pipeline," [Online]. Available: https://www.ibm.com/topics/data-pipeline. [Accessed 24/4/2024].

[7] "AWS Amazon Data Pipeline," [Online]. Available: https://aws.amazon.com/what-is/data-pipeline/. [Accessed 24/4/2024].

[8] "Snowflake ETL Pipeline," [Online]. Available: https://www.snowflake.com/guides/etl-pipeline/. [Accessed 12/5/2024].

[9] "IBM Data Quality," [Online]. Available: https://www.ibm.com/topics/data-quality. [Accessed 24/4/2024].

[10] M. Rashid and M. Torchiano, "A systematic literature review of open data quality in practice," 2016. Available: https://core.ac.uk/download/pdf/234911895.pdf.

[11] P. Oliveira and F. H. P. Rodrigues, "A Formal Definition of Data Quality Problems," in *Proceedings of the 2005 International Conference on Information Quality (MIT IQ Conference)*, Cambridge, 2005. Available: https://www.researchgate.net/publication/220918803.

[12] W. Kim, B.-J. Choi, E.-K. Hong and S.-K. a. L. D. Kim, "A Taxonomy of Dirty Data," *Data Mining and Knowledge Discovery*, vol. 7, pp. 81-99, 2003. Available: https://doi.org/10.1023/A:1021564703268.

[13] A. R. Munappy, J. Bosch, H. H. Olsson and T. J. Wang, "Modelling Data Pipelines," in *46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Portoroz, 2020. Available: https://doi.org/10.1109/SEAA51224.2020.00014.

[14] V. Garousi, M. Felderer and M. V. Mäntylä, "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering," 2018. Available: https://doi.org/10.1016/j.infsof.2018.09.006.

[15] "EBSCO Discovery Service," [Online]. Available: https://www.ebsco.com/products/ebsco-discovery-service. [Accessed 24/4/2024].

[16] "EBSCO Publishers and Partnerships," [Online]. Available: https://www.ebsco.com/publishers-partnerships/content-partnerships/full-text-licensed-databases. [Accessed 24/4/2024].

[17] R. Hallikas, "Data Extraction Table," 2024. [Online]. Available: https://figshare.com/articles/thesis/Data_Extraction_Table/25780275.

[18] A. R. Munappy, J. Bosch, H. H. Olsson and T. J. Wang, "Towards Automated Detection of Data Pipeline Faults," in *27th Asia-Pacific Software Engineering Conference (APSEC)*, Singapore, 2020. Available: https://doi.org/10.1109/APSEC51365.2020.00043.

[19] E. Aaron, "15 Common Data Quality Issues & 5 Tips to Solve Them," 2023. [Online]. Available: https://portable.io/learn/data-quality-issues. [Accessed 30/3/2024].

[20] F. J. d. Haro-Olmo, A. Valencia-Parra, Á. J. Varela-Vaca and J. A. Álvarez-Bermejo, "ELI: an IoT-aware big data pipeline with data curation and data quality," *PeerJ Computer Science*, vol. 9, 2023. Available: https://doi.org/10.7717%2Fpeerj-cs.1605.

[21] Y. Lee, Y. Noh and U. Lee, "Data Processing Pipeline of Short-Term Depression Detection with Large-Scale Dataset," in *International Conference on Big Data and Smart Computing (BIGCOMP)*, Jeju, 2023. Available: https://doi.org/10.1109/BigComp57234.2023.00095.

[22] V. Golendukhina, H. Foidl, M. Felderer and R. Ramler, "Preliminary findings on the occurrence and causes of data smells in a real-world business travel data processing pipeline," in *SEA4DQ 2022: Proceedings of the 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things*, Singapore, 2022. Available: https://doi.org/10.1145/3549037.3561275.

[23] D. Roman, A. Pultier, X. Ma and A. Ulyashin, "Data quality issues in solar panels installations: A case study," in *SEA4DQ 2022: Proceedings of the 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things*, Singapore, 2022. Available: https://doi.org/10.1145/3549037.3564120.

[24] F. J. d. Haro-Olmo, Á. Valencia-Parra, Á. J. Varela-Vaca and J. A. Álvarez-Bermejo, "Data curation in the Internet of Things: A decision model approach.," *Computational and Mathematical Methods in Statistics*, vol. 3, no. 6, p. e1191, 2021. Available: https://doi.org/10.1002/cmm4.1191.

[25] F. Pervaiz, A. Vashistha and R. Anderson, "Examining the challenges in development data pipeline," in *COMPASS '19: Proceedings of the 2nd ACM SIGCAS Conference on Computing and Sustainable Societies*, Accra, 2019. Available: https://doi.org/10.1145/3314344.3332496.

[26] I. F. Ilyas and F. Naumann, "Data Errors: Symptoms, Causes and Origins," 2022. Available: http://sites.computer.org/debull/A22mar/p4.pdf.

[27] A. R. Munappy, J. Bosch, H. H. Olsson and A. Jansson, "On the Impact of ML use cases on Industrial Data Pipelines," in *28th Asia-Pacific Software Engineering Conference (APSEC)*, Taiwan, 2021. Available: https://doi.org/10.1109/APSEC53868.2021.00053.

[28] Á. Valencia Parra, "On the enhancement of Big Data Pipelines through Data Preparation, Data Quality, and the distribution of Optimisation Problems," 2022. Available: https://dialnet.unirioja.es/servlet/tesis?codigo=309411.

[29] Á. Valencia Parra, "Analysis of Big Data Architectures and Pipelines: Challenges and Opportunities," 2019. Available: https://idus.us.es/handle/11441/114795.

[30] J. Merino, N. Moretti, M. Herrera, P. Woodall and A. K. Parlikad, "Quality-Aware Data Pipelines for Digital Twins," 2023. Available: http://dx.doi.org/10.2139/ssrn.4618449.

[31] S. Jäger, A. Allhorn and F. Bießmann, "A benchmark for data imputation methods," *Front Big Data*, vol. 4, 2021. Available: https://doi.org/10.3389/fdata.2021.693674.

[32] F. Biessmann, J. Golebiowski, T. Rukat, D. Lange and P. Schmidt, "Automated data validation in machine learning systems," *IEEE Data Engineering Bulletin*, 2021. Available: https://www.amazon.science/publications/automated-data-validation-in-machine-learning-systems.

[33] C. A. Ardagna, P. Ceravolo and E. Damiani, "Big data analytics as-a-service: Issues and challenges," in *IEEE International Conference on Big Data*, Washington DC, 2016. Available: https://doi.org/10.1109/BigData.2016.7841029.

[34] I. Taneja, "Data Pipeline and its challenges 1-4," 2022. [Online]. Available: https://www.youtube.com/watch?v=5SldUWXIfH0. [Accessed 25/3/2024].

[35] E. Wallace, "Top Data Pipeline Challenges and What Companies Need to Fix Them," 2023. [Online]. Available: https://www.clouddatainsights.com/top-data-pipeline-challenges-and-what-companies-need-to-fix-them/. [Accessed 27/3/2024].

[36] W. Yaddow, "How Modern Data Observability Tools Can Confront Evolving Data Pipeline Challenges," 2024. [Online]. Available: https://medium.com/@wyaddow/how-modern-data-observability-tools-can-confront-evolving-data-pipeline-challenges-part-1-877d5226a8fe. [Accessed 30/3/2024].

[37] F. Biessmann, T. Rukat, P. Schmidt, P. Naidu, S. Schelter, A. Taptunov, D. Lange and D. Salinas, "DataWig: Missing value imputation for tables," *Journal of Machine Learning Research*, vol. 20, no. 175, pp. 1-6, 2019. Available: https://jmlr.org/papers/v20/18-753.html.

[38] T. Mattila, "Developing a process model for incorporating data quality validation into data pipelines," 2023. Available: https://lutpub.lut.fi/handle/10024/166670.

[39] H. Zhang, G. Ruan, R. Giesting and L. Miller, "Engineering Large Wearable Sensor Data towards Digital Measures," in *IEEE 8th International Conference on Big Data Analytics (ICBDA)*, Harbin, 2023. Available: https://doi.org/10.1109/ICBDA57405.2023.10104923.

[40] R. Wrembel, "Still Open Problems in Data Warehouse and Data Lake Research: Extended abstract," in *Eighth International Conference on Social Network Analysis, Management and Security (SNAMS)*, Gandia, 2021. Available: https://doi.org/10.1109/SNAMS53716.2021.9732098.

[41] X. Huang, A. Banerjee, C.-C. Chen, C. Huang, T. Y. Chuang, A. Srivastava and R. Cheveresan, "Challenges and solutions to build a data pipeline to identify anomalies in enterprise system performance," *Data Centric AI at 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021. Available: https://doi.org/10.48550/arXiv.2112.08940.

[42] F. Pervaiz, "Understanding challenges in the data pipeline for development data," 2019. Available: https://digital.lib.washington.edu/researchworks/handle/1773/44145.

[43] P. Dazzeo, "Uncertainty propagation in experimental data pipelines," 2022. Available: https://www.politesi.polimi.it/handle/10589/211415.

[44] T. Osborn, "8 Data Quality Issues and How to Solve Them," 2023. [Online]. Available: https://www.montecarlodata.com/blog-8-data-quality-issues. [Accessed 27/3/2024].

[45] H. Zou and K. Xiang, "A Novel Rigorous Measurement Model for Big Data Quality Characteristics," in *IEEE International Conference on Big Data*, Osaka, 2022. Available: https://doi.org/10.1109/BigData55660.2022.10020564.

[46] A. R. Munappy, J. Bosch and H. H. Olsson, "On the Trade-off Between Robustness and Complexity in Data Pipelines," in *Quality of Information and Communications Technology*, 2021. Available: https://doi.org/10.1007/978-3-030-85347-1_29.

[47] S. Loetpipatwanich and P. Vichitthamaros, "Sakdas: A Python Package for Data Profiling and Data Quality Auditing," in *1st International Conference on Big Data Analytics and Practices (IBDAP)*, Bangkok, 2020. Available: https://doi.org/10.1109/IBDAP50342.2020.9245455.

[48] R. Rahman, K. C. Roy and S. Hasan, "Understanding Network Wide Hurricane Evacuation Traffic Pattern from Large-scale Traffic Detector Data," in *IEEE International Intelligent Transportation Systems Conference (ITSC)*, Indianapolis, 2021. Available: https://doi.org/10.1109/ITSC48978.2021.9564480.

[49] J.-D. Therrien, N. Nicolaï and P. A. Vanrolleghem, "A critical review of the data pipeline: how wastewater system operation flows from data to intelligence," *Water Science & Technology*, vol. 82, no. 12, pp. 2613-2634, 2020. Available: https://doi.org/10.2166/wst.2020.393.

[50] S. Abdellaoui, F. Nader and R. Chalal, "QDflows: A System Driven by Knowledge Bases for Designing Quality-Aware Dataflows," *Journal of Data and Information Quality*, vol. 8, no. 3-4, pp. 1-39, 2017. Available: https://doi.org/10.1145/3064173.

[51] H. Foidl and M. Felderer, "Risk-based data validation in machine learning-based software systems," *MaLTeSQuE 2019: Proceedings of the 3rd ACM SIGSOFT International Workshop on Machine Learning Techniques for Software Quality Evaluation*, pp. 13-18, 2019. Available: https://doi.org/10.1145/3340482.3342743.

[52] S. Deshpande, "Introducing AWS Glue Data Quality for ETL Pipelines | Amazon Web Services," 2023. [Online]. Available: https://www.youtube.com/watch?v=m4OKjfgsZ00. [Accessed 24/3/2024].

[53] N. Taverna, J. Weers, S. Porse, A. Anderson, Z. Frone and E. Holt, "An Update on the Geothermal Data Repository's Data Standards and Pipelines: Geospatial Data and Distributed Acoustic Sensing Data," in *Geothermal Rising Conference*, Reno, 2023. Available: https://www.nrel.gov/docs/fy24osti/86935.pdf.

[54] P. Rahman, A. Nandi and C. Hebert, "Amplifying domain expertise in clinical data pipelines," *JMIR Medical Informatics*, vol. 8, no. 11, 2020. Available: https://doi.org/10.2196/19612.

[55] K. Hu, "5 Common Data Quality Challenges (and How to Solve Them)," 2023. [Online]. Available: https://www.metaplane.dev/blog/data-quality-challenges. [Accessed 30/3/2024].

[56] H. Huang, "Four Challenges for ML data pipeline," 2023. [Online]. Available: https://ubuntu.com/blog/four-challenges-for-ml-data-pipeline. [Accessed 30/3/2024].

[57] J. Byabazaire, G. O'Hare and D. Delaney, "Data quality and trust: Review of challenges and opportunities for data sharing in iot," *Electronics,* vol. 9, no. 12, 2020. Available: https://doi.org/10.3390/electronics9122083.

[58] E. Kauhanen, "Continuous data quality validation in railway operation domain," 2023. Available: https://aaltodoc.aalto.fi/items/6a786e81-fc04-4cab-83fc-1b0a999f6ffc.

[59] R. Tardio, A. Mate and J. Trujillo, "An iterative methodology for defining big data analytics architectures," *IEEE Access,* vol. 8, pp. 210597-210616, 2020. Available: https://doi.org/10.1109/ACCESS.2020.3039455.

[60] I. Sacolick, "6 ways to avoid and reduce data debt," 2023. [Online]. Available: https://www.infoworld.com/article/3691789/6-ways-to-avoid-and-reduce-data-debt.html. [Accessed 8/3/2024].

[61] Marketing Evolution, "Marketing Evolution Announces New Industry Standard for Data Quality Assurance," 2019. [Online]. Available: https://www.marketingevolution.com/knowledge-center/marketing-evolution-announces-new-industry-standard-for-data-quality-assurance. [Accessed 8/3/2024].

[62] A. Immonen, P. Paakkonen and E. Ovaska, "Evaluating the Quality of Social Media Data in Big Data Architecture," *IEEE Access,* vol. 3, pp. 2028-2043, 2015. Available: https://doi.org/10.1109/ACCESS.2015.2490723.

[63] B. Klievink, E. Van Stijn, D. Hesketh, H. Aldewereld, S. Overbeek, F. Heijmann and Y.-H. Tan, "Enhancing Visibility in International Supply Chains: The Data Pipeline Concept," *International Journal of Electronic Government Research,* vol. 8, no. 4, 2012. Available: https://doi.org/10.4018/jegr.2012100102.

[64] A. Livera, M. Theristis, E. Koumpli, G. Makrides, J. Stein and G. E. Georghiou, "Guidelines for ensuring data quality for photovoltaic system performance assessment and monitoring.," in *37th European Photovoltaic Solar Energy Conference and Exhibition*, 2020. Available: https://www.researchgate.net/publication/344269813.

[65] S. Sadiq, T. Dasu, X. L. Dong, J. Freire, I. F. Ilyas, S. Link, M. J. Miller, F. Naumann, X. Zhou and D. Srivastava, "Data quality: The role of empiricism," *ACM SIGMOD Record,* vol. 46, no. 4, pp. 35-46, 2018. Available: https://doi.org/10.1145/3186549.3186559.

[66] S. May and D. Fuller, "Building Data Quality pipelines with Apache Spark and Delta Lake," 2021. [Online]. Available: https://www.youtube.com/watch?v=_ra67du8BMY. [Accessed 26/3/2024].

[67] S. Pulkka, "The Modernization Process of a Data Pipeline," 2023. Available: https://www.doria.fi/handle/10024/187373.

[68] K. Scott, "Improve Your Data Quality - Building Data Pipelines with Bad Data in Mind," 2021. [Online]. Available: https://www.youtube.com/watch?v=rEBRgiEChQI. [Accessed 26/3/2024].

[69] W. Yaddow, "Developing a Robust Data Quality Strategy for Your Data Pipeline Workflows," 2023. [Online]. Available: https://www.eckerson.com/articles/developing-a-robust-data-quality-strategy-for-your-data-pipeline-workflows. [Accessed 30/3/2024].

[70] S. Redyuk, Z. Kaoudi, V. Markl and S. Schelter, "Automating Data Quality Validation for Dynamic Data Ingestion.," in *International Conference on Extending Database Technology*, Edinburgh, 2021. Available: https://doi.org/10.5441/002%2Fedbt.2021.07.

[71] E. J. Husom, S. Tverdal, A. Goknil and S. Sen, "UDAVA: An unsupervised learning pipeline for sensor data validation in manufacturing," in *CAIN '22: Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, Pittsburgh, 2022. Available: https://doi.org/10.1145/3522664.3528603.

[72] Vijay, "Databricks Zero to Hero! - Session 2 | Data Pipeline to Data Lake | Challenges," 2021. [Online]. Available: https://www.youtube.com/watch?v=W8FCirJB5C0. [Accessed 26/3/2024].

[73] S. Bail, "Building a robust data pipeline with dbt, Airflow, and Great Expectations," 2020. [Online]. Available: https://www.youtube.com/watch?v=9iN6iw7Lamo. [Accessed 25/3/2024].

[74] K. Venkatram and M. A. Geetha, "Ingenious Framework For Resilient And Reliable Data Pipeline," *Natural Volatiles & Essential Oils*, vol. 8, no. 5, 2021. Available: https://www.nveo.org/index.php/journal/article/view/2953.

[75] P. Somasundaram, "Improving Real-Time Job Monitoring for Data Pipelines in The Context of Cloud Analytics," *International Journal of Computer Engineering & Technology*, vol. 14, no. 3, pp. 34-42, 2023. Available: https://www.researchgate.net/publication/374418941.

[76] P. E. Clayson, S. A. Baldwin, H. A. Rocha and M. J. Larson, "The data-processing multiverse of event-related potentials (ERPs): A roadmap for the optimization and standardization of ERP processing and reduction pipelines," *NeuroImage*, vol. 245, 2021. Available: https://doi.org/10.1016/j.neuroimage.2021.118712.

[77] I. Lee, "Big data: Dimensions, evolution, impacts, and challenges," *Business Horizons*, vol. 60, no. 3, pp. 293-303, 2017. Available: https://doi.org/10.1016/j.bushor.2017.01.004.

[78] J. Byabazaire, G. M. O'Hare and D. T. Delaney, "End-to-End Data Quality Assessment Using Trust for Data Shared IoT Deployments," *IEEE Sensors Journal*, vol. 22, no. 20, 2022. Available: https://doi.org/10.1109/JSEN.2022.3203853.

[79] D. Ovens, "Pipeline Monitoring | How to Start Monitoring Data Health in Palantir Foundry," 2022. [Online]. Available: https://www.youtube.com/watch?v=8aBPbQgqU5U. [Accessed 25/3/2024].

[80] S. Valarezo, R. Yackel, G. Firican and D. Firican, "Data Observability vs. Data Quality: A Comprehensive Discussion," 2023. [Online]. Available: https://www.lightsondata.com/data-observability-vs-data-quality-a-comprehensive-discussion/. [Accessed 30/3/2024].

[81] S. Ryza, "Data Quality as part of the Data Pipeline," 2023. [Online]. Available: https://www.youtube.com/watch?v=6BPN7TnORIc. [Accessed 24/3/2024].

[82] Y. Yao, L. Wu, B. Xie and L. Lei, "A two-stage data quality improvement strategy for deep neural networks in fault severity estimation.," *Mechanical Systems and Signal Processing*, vol. 200, 2023. Available: https://doi.org/10.1016/j.ymssp.2023.110588.

[83] D. Tu, Y. He, W. Cui, S. Ge, H. Zhang, S. Han, D. Zhang and S. Chaudhuri, "Auto-Validate by-History: Auto-Program Data Quality Constraints to Validate Recurring Data Pipelines," in *KDD '23: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, New York, 2023. Available: https://doi.org/10.1145/3580305.3599776.

[84] J. Song and Y. He, "Auto-validate: Unsupervised data validation using data-domain patterns inferred from data lakes," in *SIGMOD '21: Proceedings of the 2021 International Conference on Management of Data*, Xi'an, 2021. Available: https://doi.org/10.1145/3448016.3457250.

[85] M. Armbrust and P. Lappas, "Delta Live Tables A to Z: Best Practices for Modern Data Pipelines," 2023. [Online]. Available: https://www.youtube.com/watch?v=PIFL7W3DmaY. [Accessed 25/3/2024].

[86] K. Raval, "5 Challenges That Data Pipelines Must Solve," 2023. [Online]. Available: https://www.linkedin.com/pulse/5-challenges-data-pipelines-must-solve-datatech-vibe-s3fwf/?trk=public_post_main-feed-card_feed-article-content. [Accessed 27/3/2024].

[87] M. Ziemann, Y. Eren and A. El-Osta, "Gene name errors are widespread in the scientific literature," *Genome Biology*, vol. 17, no. 177, 2016. Available: https://doi.org/10.1186/s13059-016-1044-7.

[88] R. Choudhary, "Five Ways Your Data Pipelines Are Ruining Your Data Quality," 2021. [Online]. Available: https://www.acceldata.io/blog/five-ways-data-pipelines-ruining-data-quality. [Accessed 27/3/2024].

[89] J. Hilleary, "Cloud Data Pipeline Leaks: Challenge of Data Quality in the Cloud," 2022. [Online]. Available: https://firsteigen.com/blog/cloud-data-pipeline-leaks-challenge-of-data-quality-in-the-cloud/. [Accessed 27/3/2024].

[90] J. Byabazaire, G. M. P. O'Hare, R. Collier and D. Delaney, "IoT Data Quality Assessment Framework Using Adaptive Weighted Estimation Fusion," *Sensors*, vol. 23, no. 13, p. 5993, 2023. Available: https://doi.org/10.3390/s23135993.

[91] J. Merino, X. Xie, A. K. Parlikad, I. Lewis and D. McFarlane, "Impact of data quality in real-time big data systems," 2020. Available: https://doi.org/10.17863/CAM.59426.

[92] "Google TensorFlow Data Validation Guide," [Online]. Available: https://www.tensorflow.org/tfx/guide/tfdv. [Accessed 24/4/2024].

[93] "Amazon Deequ Library for Data Validation," [Online]. Available: https://github.com/awslabs/deequ. [Accessed 25/4/2024].

[94] J. R. Minnaar, "Developing a framework for identifying and assessing data quality issues in asset management decision-making," 2015. Available: http://dx.doi.org/10.1504/IJIQ.2007.013378.

[95] T. Catarci, M. Scannapieco, M. Console and C. Demetrescu, "My (fair) big data," in *IEEE International Conference on Big Data (Big Data)*, Boston, 2017. Available: https://doi.org/10.1109/BigData.2017.8258267.

[96] D. Bhardwaj, "Measurement Framework for Assessing Quality of Big Data (MEGA) in Big Data Pipelines," 2021. Available: https://spectrum.library.concordia.ca/id/eprint/988957/.

[97] A. Voropaeva, "Developing a Framework for Enhanced Data Pipeline Quality Management System," 2022. Available: https://www.theseus.fi/handle/10024/752702.

[98] H. Nawaz, "How to test your Data Pipelines with Great Expectations," 2023. [Online]. Available: https://www.youtube.com/watch?v=7UQ91Ib7PtU. [Accessed 24/3/2024].

[99] "Decision Model and Notation," OMG, [Online]. Available: https://www.omg.org/spec/DMN. [Accessed 24/4/2024].

[100] Z. Abedjan, L. Golab, F. Naumann and T. Papenbrock, "Synthesis Lectures on Data Management," in *Data Profiling*, 2018, pp. 1-154.Available: https://doi.org/10.1007/978-3-031-01865-7.

[101] Á. Valencia-Parra, L. Parody, Á. J. Varela-Vaca, I. Caballero and M. T. Gómez-López, "DMN4DQ: When data quality meets DMN," *Decision Support Systems*, vol. 141, 2021. Available: https://doi.org/10.1016/j.dss.2020.113450.

[102] S. Kandel, A. Paepcke, J. Hellerstein and J. Heer, "Wrangler: interactive visual specification of data transformation scripts," in *CHI '11: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, 2011. Available: https://doi.org/10.1145/1978942.1979444.

[103] V. Raman and J. M. Hellerstein, "Potter's Wheel: An Interactive Data Cleaning System," in *Proceedings of the 27th VLDB Conference*, Rome, 2001. Available: https://www.researchgate.net/publication/2380270.

[104] "Telltale: Netflix Application Monitoring Simplified," Netflix Technology Blog, 2020. [Online]. Available: https://netflixtechblog.com/telltale-netflix-application-monitoring-simplified-5c08bfa780ba. [Accessed 24/4/2024].

[105] "What is dbt?," dbt, [Online]. Available: https://docs.getdbt.com/docs/introduction. [Accessed 24/4/2024].

[106] B. Rogojan, "What Is DBT and Why Is It So Popular - Intro To Data Infrastructure Part 3," 2022. [Online]. Available: https://www.youtube.com/watch?v=8FZZivIfJVo. [Accessed 25/3/2024].

[107] W. Yaddow, "Who Is Responsible for Data Quality in Data Pipeline Projects?," 2023. [Online]. Available: https://tdan.com/who-is-responsible-for-data-quality-in-data-pipeline-projects/31253. [Accessed 30/3/2024].

[108] D. Faragó, "A High Quality Data Pipeline for Reasonable-Scale Machine Learning," *Softwaretechnik-Trends*, vol. 42, no. 4, 2022. Available: https://dl.gi.de/items/59eae342-d94b-4a38-91df-07624f7a0efe.

[109] R. Challen, "dtrackr: An R package for tracking the provenance of data," *Journal of Open Source Software*, vol. 7, no. 80, p. 4707, 2022. Available: https://doi.org/10.21105/joss.04707.

[110] T. Sofer, "CI/CD for Data - how to enhance data quality while increasing data engineering velocity," 2022. [Online]. Available: https://www.youtube.com/watch?v=dHpOFJZ3Kvg. [Accessed 26/3/2024].

[111] M. Asghari, D. Sierra-Sosa and A. S. Elmaghraby, "A topic modeling framework for spatio-temporal information management.," *Information Processing & Management*, vol. 27, no. 6, 2020. Available: https://doi.org/10.1016/j.ipm.2020.102340.

[112] B. Moses, "Good data pipeline, “bad data” problem," 2024. [Online]. Available: https://www.youtube.com/watch?v=xRmPKtKkCWM. [Accessed 24/3/2024].

[113] A. Sopan and K. Berlin, "AI Total: Analyzing Security ML Models with Imperfect Data in Production," 2021. Available: https://doi.org/10.48550/arXiv.2110.07028.

[114] I. Poloskei, "Data engineering case-study in digitalized manufacturing," in *IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Herl'any, 2021. Available: https://doi.org/10.1109/SAMI50585.2021.9378691.

[115] P. Hanrahan, C. Stolte and J. Mackinlay, "Application, Selecting a Visual Analytics". Available: https://www.tableau.com/pt-br/whitepapers/selecting-visual-analytics-application.

[116] S. K. Card, J. D. Mackinlay and B. Shneiderman, "Using vision to think," in *Readings in Information Visualization: Using Vision To Think*, 1999, pp. 579-581.Available: https://www.researchgate.net/publication/220691172.

[117] H. Ma, "Google Refine – http://code.google.com/p/google-refine/," *Technical Services Quarterly*, vol. 29, no. 3, pp. 242-243, 2012. Available: https://doi.org/10.1080/07317131.2012.682016.

[118] R. Orduz, "You Are What You Eat: Why Data Quality Matters for Machine Learning," Great Expectations, 2022. [Online]. Available: https://greatexpectations.io/blog/why-data-quality-matters-for-machine-learning. [Accessed 24/4/2024].

[119] C. G. Northcutt, L. Jiang and I. L. Chuang, "Confident Learning: Estimating Uncertainty in Dataset Labels," *Journal of Artificial Intelligence Research*, vol. 70, 2021. Available: https://doi.org/10.1613/jair.1.12125.

[120] A. Suleykin and P. Panfilov, "On Big Data-Driven Digital Ecosystem Framework for Railway Reporting," in *Proceedings of the 31st International DAAAM Symposium*, 2020, pp. 0499-0509.Available: http://dx.doi.org/10.2507/31st.daaam.proceedings.070.

License

Non-exclusive licence to reproduce thesis and make thesis public.

I, **Rain Hallikas,**

herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

Multivocal Literature Review on Data Quality Challenges in Data Pipelines,

supervised by Dietmar Alfred Paul Kurt Pfahl and Mario Ezequiel Scott.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Rain Hallikas

14/05/2024