A Systematic Review of Data Quality in CPS and IoT for Industry 4.0

ARDA GOKNIL, PHU NGUYEN, and SAGAR SEN, SINTEF, Norway
DIMITRA POLITAKI and HARRIS NIAVIS, Inlecom Innovation, Greece
KARL JOHN PEDERSEN, ABDILLAH SUYUTHI, and ABHILASH ANAND, DNV, Norway
AMINA ZIEGENBEIN, PTW TU Darmstadt, Germany

The Internet of Things (IoT) and Cyber-Physical Systems (CPS) are the backbones of Industry 4.0, where data quality is crucial for decision support. Data quality in these systems can deteriorate due to sensor failures or uncertain operating environments. Our objective is to summarize and assess the research efforts that address data quality in data-centric CPS/IoT industrial applications. We systematically review the state-of-the-art data quality techniques for CPS and IoT in Industry 4.0 through a systematic literature review (SLR) study. We pose three research questions, define selection and exclusion criteria for primary studies, and extract and synthesize data from these studies to answer our research questions. Our most significant results are (i) the list of data quality issues, their sources, and application domains, (ii) the best practices and metrics for managing data quality, (iii) the software engineering solutions employed to manage data quality, and (iv) the state of the data quality techniques (data repair, cleaning, and monitoring) in the application domains. The results of our SLR can help researchers obtain an overview of existing data quality issues, techniques, metrics, and best practices. We suggest research directions that require attention from the research community for follow-up work.

CCS Concepts: • **Software and its engineering** → **Embedded software**; *Layered systems*; • **Information systems** → **Database utilities and tools**; *Data compression*; *Data encryption*; **Information lifecycle management**; **Data analytics**; **Online analytical processing**; **Process control systems**; **Computing platforms**; • **Computer systems organization** → **Sensors and actuators**; **Embedded software**; *Sensor networks.*

Additional Key Words and Phrases: data quality, IoT, CPS, Industry 4.0, systematic review

1 INTRODUCTION

The Internet of Things (IoT) and Cyber-Physical Systems (CPS) are among the significant driving forces behind Industry 4.0 [153], in particular smart manufacturing [82]. They facilitate data acquisition from physical sensors and devices on an unprecedented scale and employ Artificial Intelligence (AI) techniques, e.g., Machine Learning (ML), to exploit the massive interconnection and large volumes of data. AI-enabled CPS/IoT systems improve decision-making and perform predictive maintenance (e.g., tool wear and product defect prediction in the manufacturing domain) for industrial processes in Industry 4.0. The quality and continuity of data are the bottlenecks for these systems. Many things may cause data quality to decline. For instance, CPS/IoT systems may encounter sensor flaws and failures (corrupted sensor measurements) due to various problems, such as electromagnetic interference, packet loss, and signal processing faults. The faith and reliance on these Industry 4.0 systems are diminished by poor data quality. Furthermore, the growing neglect of data quality leads to the accumulation of dark data (unstructured, untagged, and untapped data that has not yet been analyzed) [74]

Authors' addresses: Arda Goknil; Phu Nguyen; Sagar Sen, firstname.lastname@sintef.no, SINTEF, Oslo, Norway; Dimitra Politaki; Harris Niavis, Inlecom Innovation, Athens, Greece; Karl John Pedersen; Abdillah Suyuthi; Abhilash Anand, DNV, Oslo, Norway; Amina Ziegenbein, PTW TU Darmstadt, Darmstadt, Germany.

0360-0300/2023/4-ART $15.00
https://doi.org/10.1145/3593043

and the impregnation of biases [55]. It warrants the need for a detailed analysis of data quality problems/issues and data quality management techniques (in short, data quality techniques), i.e., techniques improving and maintaining data quality, that can run with CPS and IoT in various scenarios, which is the focus of this paper.

Addressing data quality issues/problems is not a new research idea. For various reasons, researchers from different fields have already provided different interpretations of data quality and disconnected data quality solutions. In the realm of relational databases, the notion of data quality concerns the normalization of data [79]. Most data quality issues in signal processing refer to a signal/noise ratio. The data science community has recently provided numerous tools and methods to "clean" data before feeding it into large ML pipelines. The importance of data quality is "the elephant in the room" for CPS and IoT, but improving data quality for them is still challenging and deserves special attention. First, sensor measurements are often corrupted or have missing values due to several (unpredictable) reasons (e.g., electromagnetic interference, packet loss, or signal processing faults). Second, CPS/IoT data often endure a long journey on the edge-cloud continuum: (i) sensor data obtained from monitoring industrial processes is consumed by a rugged industrial computer (a programmable logic controller - PLC) to control actuators; (ii) it is transferred to an edge device over wired/wireless communication channels using industrial communication protocols (e.g., NMEA [134], Bluetooth); and (iii) it is aggregated on edge to be transferred to the cloud using protocols (e.g., REST [135], RPC [121]). CPS/IoT systems need to detect and manage data quality issues (e.g., erroneous values, missing values, noise, data drift) at different stages of this journey and preserve data continuity on the edge-cloud continuum.

Although several surveys and Systematic Literature Reviews (SLR) study and classify data quality research for CPS and IoT (see Table 1 for a summary), they do not provide a detailed account and unified analysis of data quality research based on the needs and problems (data quality definitions, issues, and dimensions), the solutions (data quality techniques), and the technological/implementation context (software engineering techniques used for improving data quality). For instance, Zhang et al. [156] and Liu et al. [110] study the literature on data quality based on quality issues, dimensions, and measures but exclude the data quality techniques (e.g., data repair) and their solution domain (i.e., the abstract environment where the data quality technique is developed). The scope of the SLR conducted by Alwan et al. [58] is limited to the data quality challenges and approaches for smart cities. We answer three main research questions (ten sub-questions) to address data quality research for theoretical and practical implications in a much broader scope for Industry 4.0.

RQ1: *What is data quality for CPS and IoT in Industry 4.0?*

RQ2: *What data quality techniques are used for CPS and IoT in Industry 4.0?*

RQ3: *What software engineering solutions are used for data quality for CPS and IoT in Industry 4.0?*

We implement a typical four-step SLR process [104, 130, 150]: (i) the definition of research questions, (ii) a search strategy including the selection of online repositories and search strings, (iii) inclusion and exclusion criteria, and (iv) a data synthesis and extraction procedure. The search led to *fifty-one* (51) primary studies, which we analyzed using our taxonomy of data quality in data-driven paradigms to address three research questions. We also deliver a high-level summary of data quality for CPS and IoT in Industry 4.0. Researchers can use this summary and the taxonomy to classify and compare future data quality studies.

- The main data quality issues addressed by the primary studies are outliers (isolated, erroneous values), missing values, noise in data, data timeliness (freshness), high dimensionality, data inconsistency, and data veracity. The studies do not address the implications of different computing architectures for data quality issues for CPS and IoT. There is also little research discussing the reasons for data quality issues (**RQ1**).
- Although there is a large spectrum of data quality metrics, no study reports the adoption of these metrics in industrial IoT systems as a common practice. Across primary studies, data repair techniques address missing values, data veracity, and outliers. Most of these techniques are non-AI solutions having limitations in the industrial CPS/IoT context. Most data cleaning techniques are domain agnostic and may not always

Table 1. Recent systematic literature reviews and surveys on data quality in CPS and IoT for Industry 4.0.

Studies	Karkouch et al. [100]	Wang and Wang [148]	Teh et al. [144]	Liu et al. [110]	Zhang et al. [156]	Alwan et al. [58]	This study
Year of completion	2016	2019	2020	2020	2021	2022	2022
Systematic review?	✗	✗	✓	✓	✗	✓	✓
Number of databases used	*NA*	*NA*	3	6	*NA*	4	4
Manual search conducted?	✓	✓	✗	✗	✓	✗	✓
Number of primary studies	14	*NA*	57	45	*NA*	60	51
Time frame of primary studies	12-16	*NA*	96-18	12-18	*NA*	14-20	11-21
Data quality (DQ) focused?	✓	partially	✓	✓	✓	✓	✓
DQ classification schema?	implicit	✗	implicit	implicit	✗	✗	✓
IoT focused?	✓	partially	✓	✓	✓	✓	✓
IoT architecture discussed?	✓	✗	✓	✗	✓	✗	✓
Data security discussed?	✗	✗	✗	✗	✓	✗	✓

detect and clean domain-specific data quality issues. The evaluation metrics are mainly used to assess the impact of the data quality techniques on the performance of predictive analytics (**RQ2**).

- The existing data quality techniques address only particular quality management scenarios (online or offline). They are not deployed on different IoT reference architecture layers for diverse scenarios depending on the needs of the targeted IoT system. The programming languages, technologies, and models used to manage data quality are highly subject to the solution domain (e.g., ML, data mining, semantic web). For instance, Python is almost the de-facto programming language in the studies providing ML-based solutions. There is no direct relation between the database technologies and the data quality techniques (**RQ3**).

The paper is structured as follows. In Section 2, we present our Systematic Literature Review (SLR) approach. Section 3 describes our classification schemes for the primary studies. We present the results of our SLR in Section 4. Section 6 analyzes threats to the validity of our SLR. We discuss the related work in Section 5. Section 7 concludes the paper.

2 REVIEW PROCESS

This section discusses the steps of our review process using the popular guidelines [104, 130, 150]: (a) the definition of Research Questions (RQs), (b) a search strategy (selecting repositories and search strings), and (c) study selection based on inclusion and exclusion criteria. We also provide a summary of the search results in this section.

2.1 Research Questions

This SLR answers the three Research Questions (RQ)s presented in Section **1**. We extend each one with sub-questions.

RQ1 includes three sub-RQs.

- **RQ1.1-What are the data quality issues for CPS and IoT in Industry 4.0?**
- **RQ1.2-What are the application domains for data quality research? What types of data are collected?**
- **RQ1.3-What is the trade-off between data quality and data security?**

Table 2. Inclusion criteria.

	Inclusion Criteria
IC1	The paper is written in English.
IC2	The paper addresses data quality in any aspect (directly or indirectly).
IC3	The paper is about collecting and processing data from CPS, IoT and Industry 4.0.
IC4	The paper has engineering approaches (software, ML, statistics, data optimization) for data processing.
IC5	The paper uses real (physical world) data, not simulated data.
IC6	The paper is a long paper (at least four-page double-column or six-page single column).
IC7	The paper is in a final publication stage.
IC8	The paper is not a survey, a systematic literature review, or a systematic mapping.

RQ2 has four sub-RQs.

- **RQ2.1-What are the data quality metrics for data quality monitoring?**
- **RQ2.2-What are the data repair techniques?**
- **RQ2.3-What are the data cleaning techniques?**
- **RQ2.4-How are data quality techniques evaluated?**

RQ3 has three sub-RQs.

- **RQ3.1-What programming languages and solutions are used to manage data quality?**
- **RQ3.2-What data storage solutions are used to manage data quality?**
- **RQ3.3-What IoT reference architecture layers are covered in the primary studies?**

2.2 Inclusion and Exclusion Criteria

Considering the RQs and the basis of our study, we set the inclusion and exclusion criteria to reduce bias in our search and selection approach. The primary studies must meet ALL the accompanying inclusion criteria (see Table 2).

When more than one paper described different aspects of the same approach (e.g., the approach itself, an empirical investigation, and an evaluation), we considered those papers part of the same approach. If multiple papers detailed the same approach (not different parts) in different venues, we included only the most recent one with the most description. We removed the ones not written in English, those not peer-reviewed, and those not providing much content (less than four pages in double-column format or six pages in single-column format), extended abstracts, posters, or presentations. We excluded surveys, SLRs, or systematic mapping papers. However, we discussed them in the related work section (Section 5). We included all the papers in the search results without setting any publication period.

2.3 Search and Selection Strategy

We employ two common methods to find primary studies: database search [104] and manual search (snowballing) [150].

2.3.1 Database Search. Using online inquiry components of popular publication databases is the most notable approach to scan for primary studies when directing supplemental studies [104]. We used four popular publication databases, i.e., IEEE Xplore (https://ieeexplore.ieee.org), ACM Digital Library (https://dlnext.acm.org), ScienceDirect (https://sciencedirect.com/), and Scopus (https://scopus.com), to search for potential primary studies. Scopus and ACM DL already index SpringerLink (https://www.springer.com) [145]. These databases contain peer-reviewed articles and provide advanced search capacities. We defined our search keywords by

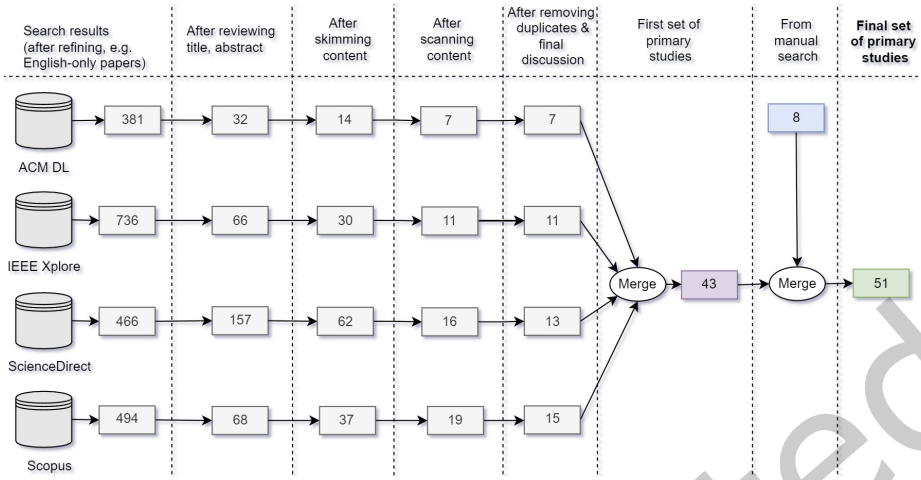

Fig. 1. Overview of the search and selection steps.

following the guidelines from [104]. The search query was adapted to fit the search engine of each publication database.

(*"Internet of Things"* **OR** *"IoT"* **OR** *"Cyber Physical Systems"* **OR** *"Industry 4.0"*)
AND (*"sensor data"* **OR** *"data completeness"* **OR** *"data quality"* **OR** *"data repair"* **OR** *"sensor calibration"*)
AND (*"machine learning"* **OR** *"AI"* **OR** *"digital twin"* **OR** *"reference model"*)

Figure 1 gives an overview of the search and selection steps. We first filtered the candidate papers based on their titles and abstracts. When the titles and abstracts were not enough to decide whether to discard or keep the papers, we continued to skim and scan through the contents of these papers. When a candidate paper appeared in more than one database, we kept it, at first, in multiple search results. Then, we consolidated the outcomes with group discussions among the authors to acquire the first set of primary studies with no duplicates.

2.3.2 Manual Search. It is unattainable to ensure that the database search covers all the primary studies. Thus, we supplemented the database search with a manual search, as suggested by Wohlin [150]. We found eight more primary studies. Please note that we kept candidate papers in doubt for further evaluation and cross-checking. Our search and selection process ended with 51 primary studies at the end of 2021 for data extraction and synthesis.

2.4 Data Synthesis and Extraction Method

This section discusses the search results and extraction methods. We included 51 primary studies in our study. We extracted related information from these studies according to our RQs (see Table 3).

There were 28 conference papers, 21 journal articles, and two workshop papers. We gathered papers from ACM International Conference on Distributed and Event-Based Systems (DEBS), ACM Symposium on Information, Computer and Communications Security (CCS), ACM Multimedia Systems Conference (MMSys), ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), IEEE International Conference on Big Data, and IEEE International Conference on Computer Communications and Networks (ICCCN). We also retrieved papers from ACM Transactions on Cyber-Physical Systems, Computers in Industry, and Information Systems. They are all well-known and credible publication venues for data quality, CPS, and IoT research. There is a growth in publication numbers from 2011 to 2021, with a sharp increase after 2018 (92% of the primary studies are published

Table 3. Data collection for each research question.

Research Question	Type of Data Extracted
RQ1	Data quality issues, their sources, data types and application domains, data quality metrics.
RQ2	Data quality techniques and their reported strengths and limitations.
RQ3	Architectures, programming technologies and databases used for data quality.

after 2018). According to this trend, we can conclude that data quality research for CPS and IoT has been popular in recent years.

3 A CLASSIFICATION SCHEMA / TAXONOMY

Kuhn defines a scientific paradigm in his book "Structure of Scientific Revolutions" [138] as "universally recognized scientific achievements that, for a time, provide model problems and solutions for a community of practitioners". We are inspired by this definition and use the term Data-driven Paradigm for systems that leverage large streams/batches of data to control, manage, and optimize processes in different industrial *sectors*. The Data-driven Paradigm is the root node of our taxonomy (as shown in Figure 2), which we extend for IoT and CPS that are data-driven to a large extent. The Data Quality Management Technique is the main entity in our taxonomy.

3.1 Data Quality Management Techniques

A data-driven paradigm contains zero to many data quality management techniques. Data quality management techniques (in short, data quality techniques) aim at improving and maintaining data quality across system components. We identify three types of techniques:

- Data Monitoring: Data is monitored to detect data quality issues such as outliers and noise.
- Data Cleaning: It is a technique that entails removing corrupt and unusable data, e.g., those affected by environmental noise or extreme operating conditions such as high temperature.
- Data Repair: It is a technique to restore data that has been lost, accidentally deleted, corrupted, or made inaccessible, e.g., by using simulation data or data from redundant sources (other sensors).

Data quality techniques can be **online** (real-time at the data source) and **offline** (for large historical datasets on the cloud). They have *zero to many* quality standards and data quality metrics.

Quality Standards provide requirements, guidelines, or characteristics used to ensure that materials, products, processes, and services serve their purpose. An example standard is a documented agreement on data representation, format, and definition. Standardization organizations define data quality standards (e.g., ISO8000 [87]).

Data Quality Metrics are the measurements by which you assess your data. They benchmark how complete, valid, accurate, timely, and consistent the data is and help differentiate between high-quality and low-quality data.

A data quality technique has *an* **Automation Level** indicating whether it is fully Automated, Manual, or supports a human operator with Assisted Decision Making.

3.2 Algorithms to Support Data Quality Techniques

A data quality technique uses zero to many Algorithms to enable automation, manual inspection, and assisted decision-making. Algorithms typically require one or more sources of Input Data and may generate zero to many Output Data. We measure algorithm performance by using zero to many Performance KPIs. For instance, input data can be time-series data from a sensor, output data can be an anomaly in the input data, and performance KPI can be the classification accuracy. We categorize algorithms that support data quality techniques as follows:

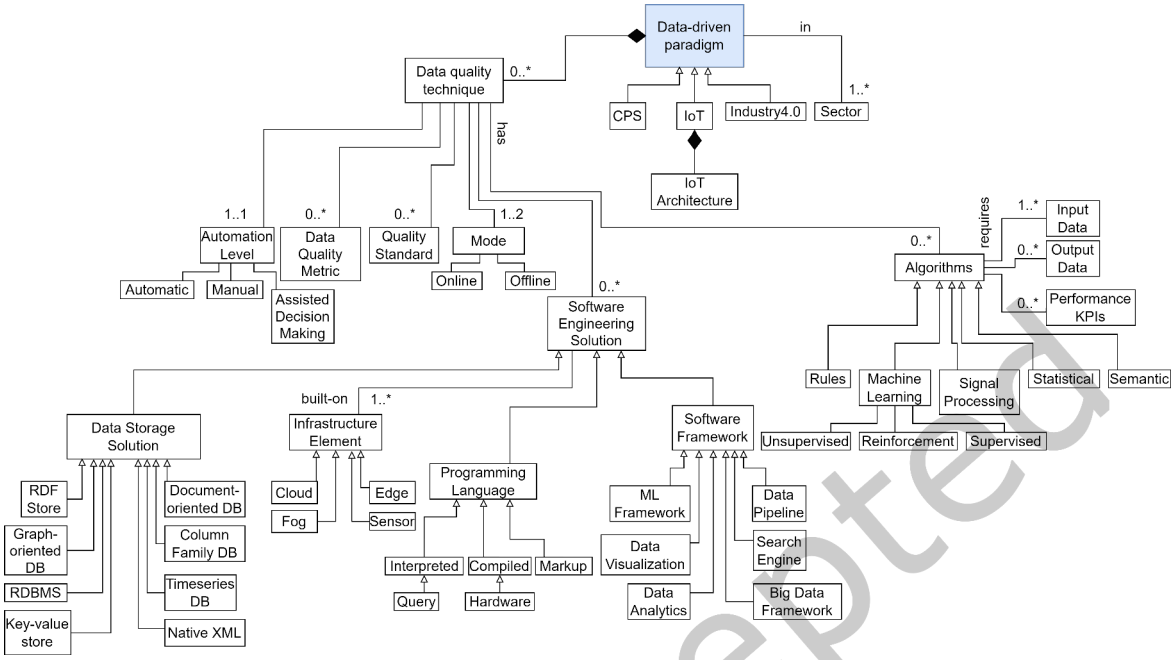

Fig. 2. Taxonomy of data quality in data-driven paradigms.

Rules are declarative constraints input data need to satisfy for high quality. They can specify what is not expected.

ML Algorithms [86] are for Supervised, Unsupervised, or Reinforcement learning. They are used to detect data quality issues or clean and repair data. Supervised learning maps an input to an output based on example input and labeled/classified output pairs. Unsupervised learning identifies patterns in input data that are neither classified nor labeled. Reinforcement learning is based on rewarding desired actions and punishing undesired ones through trial and error. **Signal Processing Algorithms** [127] analyze, modify, and synthesize sensor signals. They support the storage, compression, and reconstruction of signals, separation of information from noise, and feature extraction from signals. **Statistical Algorithms** entail the creation of a statistical model of the input data and use statistical quantities such as min, max, median, standard deviation, and quartiles on the input data to detect data quality.

3.3 Software Engineering Solutions to Support Data Quality Techniques

Software Engineering Solutions (Data Storage Technologies, Programming Language, and Software Framework) support data quality techniques. Being aware of multiple interpretations, we use the term software framework as a software engineering solution (e.g., data pipeline, big data platforms) providing generic software functionality that can be selectively changed by additional user-written code, thus providing application-specific software.

These solutions are built on Architecture/Infrastructure Elements in the **IoT architecture**, such as Sensors, Edge devices, Cloud infrastructure, and, in some cases, local Fog infrastructure. An Edge device connects sensors or data sources in a local area network. It can also link the local area network to a wide area network or the Internet. Cloud infrastructure offers virtual resources for scalable and reliable computation and storage. It is available on the Internet as Infrastructure as a Service (IaaS). Fog infrastructure consists of an IoT gateway within the local area network of Edge devices and connects them to the Cloud.

Data Storage Solutions: Figure 2 gives the list of data storage solutions in our taxonomy. Blockchain stores data in blocks chained together in chronological order. The common blockchain application is a ledger for transactions. Distributed file systems allow access to files from multiple hosts sharing via a network, making it possible for multiple users on multiple machines to share files and storage resources. A database is an organized collection of structured information, or data, usually controlled by a database management system (DBMS). DBMS can be categorized into [88]: (i) relational DBMS, (ii) document-oriented DBMS, (iii) graph-oriented DBMS, (iv) column family DBMS, (v) native XML DBMS, (vi) time series DBMS, (vii) Resource Description Framework (RDF) stores and (viii) key-value Stores.

Programming Languages: Our taxonomy covers programming languages in three categories: Interpreted, Compiled, and Markup. An interpreted language supports the execution of instructions without compiling them to machine code. Query languages are interpreted languages for searching, viewing, and changing the content of a database. Compiled languages translate source code to machine code as opposed to interpreted languages. Hardware description languages are compiled languages that support the automated analysis and simulation of electronic circuits. A Markup language is a system for annotating a document visually distinguishable from the content.

Software Frameworks: Our taxonomy classifies software frameworks as ML Frameworks, Data Pipelines, Data Visualization, Data Analytics, and Big Data Frameworks. ML Frameworks enable ML models to be developed without understanding the underlying algorithms. Data Pipelines process data in a sequence where the output from one component becomes the input for the next component. Data Visualization frameworks support the process of translating large data sets and metrics into charts, graphs, and other visuals. Search Engines help find the information by using keywords or phrases. Data Analytics frameworks enable data analysis in an organized way. A Big Data Framework is an ecosystem of different components that process, handle and store large amounts of data.

IoT Architecture: The IoT World Forum Reference Model [98] is one of the numerous IoT architectures in the literature. It supports fine-grained granularity across various layers that make up an IoT system. Many large-scale IoT systems have lately incorporated this architecture [75]. It has seven layers. **L1 Physical Devices and Controllers layer** contains sensors, edge node devices, and other devices. **L2 Connectivity layer** enables transferring data from the cloud to devices and vice-versa. **L3 Edge Computing layer** brings computation and storage closer to where data are gathered. The protocol conversion, routing to higher-layer functions, and "fast path" logic for low-latency decision-making are implemented here. **L4 Data Accumulation layer** converts sensor data in motion to data at rest. It stores the data in an easy-access format and reduces it through filtering and selective storing. **L5 Data Abstraction layer** focuses on rendering data and their storage in ways that enable performance-enhanced applications. Information interpretation occurs at **L6 Application layer**. The software interacts with **L5** and data at rest. Thus, it does not have to operate at network speeds. **L7 Collaboration and Processes layer** enables human interaction with all the other layers. A simpler IoT architecture adopted in the literature (e.g., [44]) consists of three layers: data acquisition/perception (**L1**), network (grouping **L2** and **L3**), and data service/application (grouping **L4**, **L5**, **L6**, and **L7**).

4 RESULTS

With the three research questions (RQ1, RQ2, and RQ3), we have investigated the context, application, and problem domains (data quality issues and sectors where data quality issues are addressed), solution domains (data quality techniques, data quality metrics, and how these techniques are evaluated), and implementation domains for solutions (programming languages, libraries, frameworks, and data storage techniques).

4.1 RQ1 - What is data quality for CPS and IoT in Industry 4.0?

This research question provides an overview of typical data quality issues in CPS and IoT for Industry 4.0, their sources, and the application domains for data quality research. To respond to RQ1, we address the following three sub-questions:

4.1.1 ***RQ1.1: What are the data quality issues for CPS and IoT in Industry 4.0?***

Once data is recorded by sensors, it is transformed in several stages until it arrives at the control room, where a decision is made automatically or manually. Like any other IT system, the principle of garbage-in-garbage-out is also valid here. Poor quality data may adversely affect the overall decision-making process. Therefore, data has to be trustworthy throughout the data value chain. Eliminating data quality issues (data quality problems) as early as possible, for example, when data are first captured, is far more efficient than handling data quality issues later in the data value chain [97]. This research question aims to identify data quality issues experienced in CPS and IoT for Industry 4.0 applications. We expect to gain insights into the reasons for these issues and later establish the connection among the data quality issues, solutions, methodologies, and application domains we investigate in the following questions.

The reviewed papers address wide-ranging data quality issues. They include, amongst others, outliers (isolated, erroneous values) [2, 8, 9, 12, 13, 15, 19, 28, 29, 34–36, 40, 42, 43, 45, 47, 49, 50], missing values [1–3, 6, 9, 14, 18, 19, 21, 25, 28, 32, 33, 35–39, 43–46, 50], duplicated records [9, 14, 19], noise in data [5, 19, 30, 33, 37, 42, 45, 48], data drift [14], data discontinuity [17], data imprecision [25], data timeliness (freshness) [1, 3, 10, 16, 21, 22, 26, 38, 39], high dimensionality [9, 19, 42, 43], data inconsistency [1, 3, 4, 6, 10, 25], and data veracity [6, 7, 11, 20, 23, 27, 31, 33, 38, 39]. Data quality issues are mainly addressed using data quality dimensions, i.e., attributes representing a single aspect of the data quality [147]. One example data quality issue is data inconsistency, and data consistency is the corresponding data quality dimension. Another one is missing values as the data quality issue of data completeness. Some primary studies use data quality issues and dimensions interchangeably (e.g., data freshness [1]). Data completeness refers to the degree to which all parts of the data are specified with no missing information. Data freshness implies that the sensed data are recent and no adversary replays old messages. The reader is referred to Wang et al. [149] for the definitions of all the other data quality dimensions.

Missing values are one of the most addressed data quality issues in the primary studies. It refers to cases when one variable or attribute does not have any value. Another highly addressed data quality issue is outliers, i.e., extreme values that deviate from other observations on data. We observe that the most addressed data quality issues are highly related to the types of data the most dealt with in the primary studies. Time series is the most addressed data type (see RQ1.2), and missing values and outliers are common problems for time-series data sets, e.g., due to sensor failures at high sampling frequency and network problems.

A usual CPS/IoT system includes many diversified components such as sensors, actuators, backend, Web, Cloud, and Web/mobile software. These components frequently interact on different computing architectures (edge, fog, and cloud computing) and participate in various workflows. We analyze the computing architectures presented in the primary studies and their relation to data quality issues (see RQ3.3). Here, we can briefly state that the studies in our review do not address the implications of these architectures for data quality issues and techniques for CPS and IoT. For instance, edge-native systems have limited computing and storage capabilities; ML applications require massive volumes of training data in high-dimensional feature space to ensure several samples with the combination of possible values for each feature. ML applications running at the edge need the high dimension in input data sets reduced to increase application performance. On the other hand, a crucial but one of the least addressed data quality issues in our review is high dimensionality. It refers to data sets that contain a large number of features [103].

Data veracity refers to how accurate or truthful a data set may be and answers questions like how trustworthy the data source, type, and processing are. We can distinguish it from data quality as it is sometimes considered a

data security property [20]. We investigate the relationship between data security and other quality issues in a separate research question (see RQ1.3). Most primary studies on data veracity address maliciously manipulated sensor signals. For instance, Krotofil et al. [20] assume attackers can tamper with sensors to hide actual sensor signals.

The primary studies have different classifications for some data quality issues. Noise is defined as irrelevant or meaningless data [152]. Corrales et al. [9] classify missing values, outliers, high dimensionality, and duplicate records as noise. Sanyal et al. [33] differentiate gaussian noise and outliers as distinct data quality issues. Some primary studies (e.g., [30, 45]) refer to noise as unwanted and wrong data that should be removed. Some of the studies have different interpretations of outliers and anomalies. For instance, Huru et al. [15] use terminology that maps outliers to measurement errors and anomalies to unusual events in time series data such as temperature and humidity collected from sensors placed inside a greenhouse environment. Saybani et al. [35] define anomalies as sensor faults that lead to missing values and outliers in sensed data, while Kong et al. [19] use "abnormal data" as the term for outliers. Yu et al. [48] treat anomalies as deviations different from noise, which is erroneous, out of all potential values, and shows up as a spike. Anomalies represent system failures and slightly deviate from common values but still occur inside the range. They change slowly as system failures take time to stop machine operations. Based on this definition, Yu et al. provide a noise filter that removes noise and preserves anomalies (see RQ2). Flick et al. [12] follow another definition of outliers, noise, and anomalies. Outliers cover both noise and anomalies and are observations that deviate so much from others (normal data). Noise represents the semantic boundary between normal data and true anomalies. It is a weak form of outliers, focusing on a single data point, whereas anomalies are inferred collectively from a set of data points.

Not many papers discuss the reasons for data quality issues (the reasons for data quality problems) and their impact on the proposed solutions. The most mentioned reasons are heterogeneous multiple data sources [4, 12, 30, 44, 50], sensor malfunctions [15, 31, 35], network problems (connection failures, communication delays) [16, 37, 40], high sampling frequency [19], and cyber attacks for data tampering [7, 31]. Wang et al. [44] state that dependable, raw time series (time series gathered from multiple sensors) are very likely to contain missing values, which could harm the accuracy of data analytics. They use the reason (multiple data sensors) to devise their data quality technique. To reconstruct missing data, they utilize the correlations between time series generated by sensors working together.

Flick et al. [12] discuss that data from different sources lead to several data quality issues, such as missing values, outliers, and missing or invalid time stamps. They provide a conceptual framework for data pre-processing of diverse data sources. The framework deals with different or even not existing timestamps to merge data from multiple data sources. It also deploys outlier treatment algorithms to detect outliers in multivariate data sets. Another reason for low-quality data is cyber-physical attacks. Casado-Vara et al. [7] study incidents where malicious data lead to poor data quality. They present a blockchain-based architecture to improve data security, with an edge computing layer executing a new algorithm using game theory for false data detection.

Kong et al. [19] focus on duplicated data, missing values, and outliers caused by high sampling frequency and the vast number of installed sensors. They use high sampling frequency to calculate the missing value based on the average value of the previous and next data. If there is a wide range of missing data, missing values are predicted from the data collected from other sensors observing a similar phenomenon. As seen in these few studies, the reasons for data quality issues can be crucial in devising data quality management techniques. Therefore, we need more research to investigate the dependency among data quality issues, their reasons, and data quality management techniques.

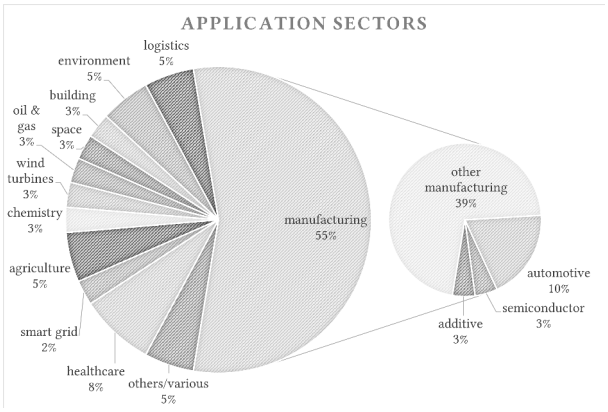

Fig. 3. Application domains of the data quality research for CPS and IoT in Industry 4.0.

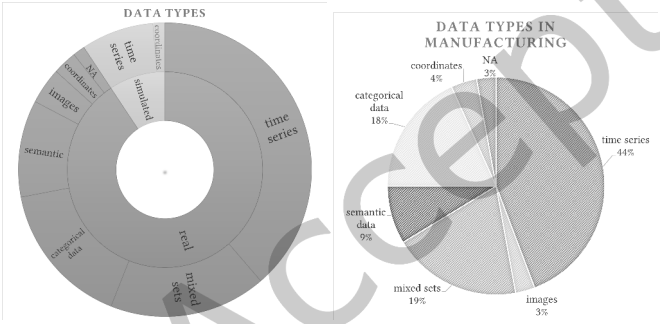

Fig. 4. Data types of the data quality research for CPS and IoT in Industry 4.0.

RQ1.1 Conclusion. Data quality research for IoT and CPS mainly addresses missing values, outliers, and data timeliness. The studies in our review do not address the implications of different computing architectures for data quality issues and techniques for CPS and IoT. There is also little research discussing the reasons for data quality issues (e.g., electromagnetic interference, high temperatures, loss of connectivity, and signal processing errors). Future research should further investigate the reasons and the implications of computing architectures for data quality issues to devise better data quality management techniques.

4.1.2 *RQ1.2: What are the application domains for data quality research? What types of data are collected in these application domains?*

Various industries are concerned with questions and research around data quality. Figure 3 presents the application domains of the primary studies in our review. A big part of the literature (55%) stems from research for manufacturing industries (e.g., automotive, semiconductor). Healthcare, environmental research, agriculture, and logistics follow manufacturing industries. The use of data analytics technologies in manufacturing industries to ensure quality is getting more and more attention to increase performance and yield, reduce costs, and optimize supply chains. It is known as manufacturing analytics and is part of Industry 4.0, where factories

evolve into self-running and healing entities by adopting new technologies such as IoT. Therefore, most primary studies on manufacturing industries propose manufacturing analytics frameworks and support data quality as a preprocessing step for data analytics input. For instance, Weiss et al. [46] provide methods for continually predicting manufactured product quality in semiconductor manufacturing. The proposed methods have a data preprocessing step to predict missing input data. We can conclude that data quality is crucial to the manufacturing business since manufacturing involves multiple sensors in harsh environments (e.g., a shop floor - the area of a factory, machine shop), which may lead to various data quality issues. Please note that the most mentioned reasons for data quality issues in RQ1.1, such as heterogeneous multiple data sources, sensor malfunctions, and network problems, are likely to occur in such environments.

The data types in the reviewed literature are manifold (see Figure 4). Time-series data dominate the data types. Most evaluations in the primary studies are conducted in a controlled research environment rather than in an industrial setting. Only six of 51 studies present an evaluation in an industrial environment (e.g., a shop floor).

RQ1.2 Conclusion. Manufacturing industries are the predominant application domain in the primary studies, while time series are the most collected data type in the application domains.

4.1.3 ***RQ1.3: What is the trade-off between data quality and data security?***

Data security is to prevent unauthorized access to data. It can be considered part of data quality (see data veracity in RQ1.1) since avoiding data corruption caused by unauthorized access improve data quality. Data quality techniques, e.g., data cleaning or repair, necessitate flexible read and write access to all data. Security problems may arise while running these techniques because data can be exchanged with other systems or used by users with different access rights. Improving data security may limit the abilities of data quality techniques and, in turn, reduce data quality.

The need for data security enforces certain restrictions on how data is accessed, stored, and analyzed. These restrictions may negatively affect overall data quality and increase computational costs. For instance, data may need to be anonymized or encrypted in a vault. Encryption techniques require different users to encrypt their data with their keys; the identical data copies of different users lead to different ciphertexts, making data deduplication impossible. Data deduplication eliminates redundant copies, significantly reduces storage capacity requirements, and ensures data consistency. Some well-adapted strategies, e.g., centralized secret keys within a dedicated entity that allow the deduplication process to decrypt data, can be employed to overcome conflicts between data quality and security [143].

We have identified only seven papers addressing both data quality and security for IoT and CPS. Four of these papers [7, 11, 20, 31] investigate how data security solutions can improve data quality (in particular, data accuracy). Krotofil et al. [20] propose a process-aware approach to detect when a sensor signal is maliciously changed. Similarly, Russel et al. [31] present a sensor data validation method that employs sensory substitution to mitigate common sensing errors and cyber-physical attacks, such as playback attacks. Two blockchain-based approaches [7, 11] support the assessment of the trustworthiness of sensor observations and false data detection to improve IoT data quality.

Only three papers [38, 39, 51] study the trade-off between data quality and security. Zellinger et al. [51] address confidentiality protection in transfer learning (an ML approach focusing on storing knowledge gained while solving one problem and applying it to a different but related problem). The idea is a module-based combination of confidentiality-preserving noise-adding methods with robust transfer learning algorithms for intelligent manufacturing applications. They discuss noise injection mechanisms achieving a good trade-off between data privacy and accuracy.

Sicari et al. [38, 39] propose a system architecture addressing data security and quality in IoT. The architecture contains three layers: *analysis*, *data annotation*, and *integration*. The Analysis layer extracts the information about the data (e.g., data source, data type, and data security and quality properties) to support other layers. Its task is to evaluate whether the input data satisfy data security and quality requirements. The layer computes a score for each security property (i.e., data confidentiality, data integrity, source privacy, and source authentication) and data quality property (i.e., data accuracy, data precision, information timeliness, and completeness). These scores inform IoT users and applications about the security and data quality levels. The Data Annotation layer annotates the data with the computed scores; the Integration layer exploits the scores about security and quality level to select the data resources for data integration. The evaluation of the proposed architecture reveals that high-security scores may lead to low-level data completeness (the number of collected values over a given time interval divided by the number of expected values). On the other hand, high-level data security positively contributes to data accuracy (the degree of similarity of a measured quantity to its actual value), precision (the degree to which further calculations return the same or similar results), and timeliness (the temporal validity of data).

Different computing architectures (e.g., edge, fog, and cloud computing) have different security risks, affecting the trade-off between data security and quality. For instance, edge applications pose particular security risks (e.g., the dependence on edge computing resources without proper security software) because IoT devices are designed for low-cost and low-power usage and are unsuitable for complex technology. One technique to mitigate these risks is to monitor all edge activity to limit data access rights of native edge applications, including data quality techniques running on edge devices. Similar to our findings in RQ1.1, none of the primary studies mentioned above discuss the implications of IoT computing architectures for the trade-off between data security and quality.

RQ1.3 Conclusion. Some primary studies focus on data quality and security separately or study how data security can improve data quality and therefore do not address the trade-off between these two. Few papers addressing the trade-off study data quality and security properties, but not how data quality and security techniques affect or limit each other on different computing architectures. We need further research on the implications of different IoT computing architectures for the trade-off between data security and quality.

4.2 RQ2 - What data quality techniques are used for CPS and IoT in Industry 4.0?

Data quality techniques include approaches and technologies identifying and correcting data quality issues. There are different interpretations of data quality techniques. Some surveys mix data cleaning and repair under the same category. Our SLR has a distinction between data repair and cleaning techniques (see Figure 2) and reports them in two sub-questions (RQ2.2 and 2.3) since they differ in how they treat data quality issues. Data monitoring (identifying quality issues in data) is a prerequisite and an integral part of data repair and cleaning. Therefore, we investigate it as a sub-activity of data repair and cleaning. To respond to RQ2, we address four sub-questions:

4.2.1 *RQ2.1: What are the data quality metrics for data quality monitoring?*

Data quality metrics are the measurements used to assess data. They benchmark how beneficial and relevant data is, and help differentiate between high-quality and low-quality data. They can be employed to certify data sources in IoT and CPS as fit or not for specific purposes. They can easily be related to data quality dimensions, i.e., the measurement attributes of data, which we can assess, interpret, and improve.

We identified 41 data quality metrics in the primary studies. Table 4 presents the data quality dimensions and the corresponding metrics with their formulas/explanations. The metrics are either based on a mathematical formula or computed by a program on structured data for an observation period. Accuracy, completeness, and validity are the most data quality dimensions addressed by the quality metrics. We could not find any metric for

Table 4. Data quality metrics in the primary studies.

DQ Dimension	Number	Formula / Description	Reference
Accuracy	M1	$X = \{x_t \mid t \in T\}$ is a time series process, and N is the number of observations. Degradation of accuracy is detected as deviation in the following properties of X. Mean $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$. Standard deviation $\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_t-\mu)^2}$. Kurtosis $K = \frac{1}{N}\frac{\sum_{i=1}^{N}(x_i-\mu)^4}{\sigma^4}$. Skewness $S_k = \frac{1}{N-1}\frac{\sum_{i=1}^{N}(x_i-\mu^3)}{\sigma^3}$. Sum of absolute values $\sum_{i=1}^{N}\lvert x_i\rvert$. Number of elements over the mean $\{x_t \in X : x_t > \mu\}$	[8]
	M2	Test campaigns using human to detect data quality faults in facility management using cameras and sensors. O is the occupancy count obtained from cameras and sensors. $O > 50$ in a 10 meter squared room indicates data error. $O > 0$ outside working hours then the sensor is frozen.	[36]
	M3	Binary logistic regression $p(x_i) = \frac{1}{1+e^{-\beta x_i}}$, where $p(x_i)$ is the probability that the data point x_i is noisy.	[9]
	M4	Accuracy score is between [0, 1] revealing proximity of values to correct (range of) values. $z_i = \frac{x_i - min(X)}{max(X)-min(X)}$	[38]
	M5	*Accuracy is computed as* $1 - V_T/N_A$*, where* V_T *is the number of tuples in a relation having one or more incorrect values and* N_A *the total number of tuples.*	[6]
Artificiality	M6	$q_{art} = 1$ if the sensor information originates from an individual IoT sensor, which is not aggregated or interpolated. $q_{art} = 0$ if data is from an unidentified information source, which aggregates information with unidentified algorithms.	[21]
Availability	M7	Percentage of the time that there is an unexpired data object provided by the source for a given observation period. $Availability = 1 - \frac{\sum_{i=1}^{n} max(0, t_i - T^{exp})}{OP}$, where, for an observation period OP, t_i is the interval between the i_{th} and the $i+1_{th}$ updates, n is the total number of objects received in OP, and T^{exp} is the expiration time.	[22]
Completeness	M8	The MD5 hash serves as a binary metric for completeness and is used to verify if the source data files are empty empty.	[32]
	M9	An *aggregate quantity*, i.e., *occupancy count* O (number of people in the facility), determines missing data, outliers, stuck values, and noisy data. For instance, if O is zero during working hours, there may be missing data. (Related to M2)	[36]
	M10	$q_{cmp} = 1 - M_{miss}/M_{exp}$, where M_{miss} is the sum of missing values, and M_{exp} is the sum of expected values.	[21]
	M11	Percentage of data fields that are associated with a proper value.	[38],
	M12	Packet Loss Rates in the NASA Turbofan dataset in order to mitigate the effects of wireless network degradation.	[41]
	M13	Completeness is quantified as $1 - T_R/N_R$, where, for a relation R, T_R is the number of tuples in R that have at least one "NULL" value and N_R is the total number of tuples in R.	[6]
Comprehensiveness	M14	It is quantified as conformance to a hierarchical model in ontological reasoning. Threshold is the only parameter class used for each data variable. It can be a value or an interval.Its class has three properties: type, upper threshold and lower threshold. If the type is a value, it is set to "0" and either upper or lower threshold is set. Otherwise, the type is set to "1" and both upper and lower thresholds are set. Abnormal data is classified based on three times threshold values.	[19]
Concordance	M15	It is quantified as the agreement between data source information and information of further independent sources. $q_{con} = \sum_{i=1}^{n} \lambda_i \cdot c(x_0, x_i)$, where n is the number of sensors, $c(x_0, x_i) \in [0, 1]$ is the share of measurements witnessing sensor observation, λ is a weight function $\lambda_i(x_0) = \frac{1}{d(x_0, x_i)}$, if $d(x_0, x_i) \neq 0$, and $d(x_a, x_b)$ is the propagation and infrastructure-based distance function between sensor locations x_a and x_b for sensors a and b	[21]
Confidentiality	M16	Data confidentiality (a binary trait) is established as a consequence of using federated learning in lieu of central data collection followed by machine learning.	[51]
Currency	M17	Currency is computed as statistical properties of data (as in M1) to observe degradation over long periods spanning several days and months.	[8]
	M18	Currency is the query spatial difference to measure object staleness in virtual objects in driving simulations.	[16]
	M19	$Currency = (1 - \frac{Age}{T^{exp}}) * e^{-Volatility}$, $T^{exp} \geq Age$, see M41 for Volatility	[22]
Privacy	M21	Score between [0, 1] for privacy. Adoption of a privacy model and related privacy policies are associated to a high score.	[38]
Redundancy	M22	Multi-collinearity to show high inter-correlations among two or more independent variables and removes attributes whose values can be trivially predicted by a multiple regression model of the other attributes hence reducing	[8]
	M23	Same approach as M3 but to classify data as redundant using logistic regression.	[9]
	M24	Same as M17 but statistical properties are used to compute redundancy.	[8]
Reputation	M25	Score between [0, 1] representing probability by which data are suitable to be included in a process providing value.	[38]
Security	M26	Score between [0, 1] for data security. Data authentication represents the need to identify users or objects authorized to access data. A score is assigned to authentication and integrity levels.	[38]
Timeliness	M27	If a measured value is within bounds of age and frequency the reward increases otherwise it is punished. Timeliness is given by $q(t) = \lvert q(t-1) - 2 * Rd(t)\rvert$, where reward $Rd(t) = \frac{\alpha^{W-1}(t-1)}{W-1} - \frac{\alpha^{W-1} + \alpha^{current}(t)}{W}$, W is the length of the sliding window, α^{W-1} denotes the number of measurements within the given interval, $\alpha^{current} \in \{0, 1\}$ is the current reward/punishment decision (1 for a measurement within the interval or 0 otherwise).	[21]
	M28	Score between [0, 1] is the extent to which data age is appropriate for the task at hand. The temporal validity of the data is defined by age and volatility. Age is a measure of how old the information is, based on when it is recorded.	[38]
Trust	M29	Trust T_i for data stream i is $T_i = w_1.Accuracy + w_2.Completeness + e$. Related to M5 and M13.	[6]
Uncertainty	M30	Shannon entropy typically used to measure amount of information in a variable. If the entropy is high then the chances of surprises is high and if the entropy is low the chances of surprises is low.	[33]
Validity	M31	Same as M17 but deviation from statistical properties are used to compute validity.	[8]
	M32	*Data Validity* from sensor data gathered every 5 minutes based on binary detection of Anomaly or not.	[35]
	M33	It is quantified as Local Outlier Factor (LOF) value. The algorithm uses k-nearest neighbor on inserted data records to instantly compute LOF value, i.e., the degree to which a record represents an outlier or an indicator of abnormality.	[47]
	M34	Percentage of filtered data for rare events in the G-APD Cherenkov Telescope (FACT).	[5]
	M35	Information Gain Metric. Furthermore, outliers are detected if no information is gained and removed.	[40]
	M36	Metric is based on minimum accepted deltas using domain/user knowledge.	[15]
	M37	It is satisfaction of consistency rules using semantic reasoning and Non-Description Logic on a sliding data window.	[4]
	M38	Sensory substitution to compute number of false positives and negatives, and corroboration of true positives and negatives.	[31]
	M39	$Validity = \wedge_{i=1}^{m} VR_i(o)$, where VR_i is ith rule in a set of m rules, for data object o. Validity for historical performance is $\frac{n_{valid}}{n}$, where n is the number of updates, and n_{valid} is the valid instances updated.	[22]
Volatility	M40	Score between [0, 1] is a measure of information instability, the frequency of change of the value for an entity attribute.	[38]
	M41	Probability of an update between the last one (time point 0) and the current time (age). $V = \int_0^{Age} f^{update}(t)dt$	[22]

data orderliness, volume, auditability, consistency, accessibility, compliance, efficiency, precision, traceability, understandability, portability, recoverability, and integrity. Data consistency among these dimensions is the level to which values of an attribute adhere to some constraints. Data validity metrics (M31-M40) subsume this definition of data consistency.

Several data quality metrics in the primary studies are numerically *bounded*. Some metrics produce a normalized score in the range $[0, 1]$ as a binary value (M6, M16), percentage (M7, M11, M34), score (M4, M21, M25, M28, M40), relative frequency (M5, M10, M13), and probability (M3, M41). The bounded metrics appear to be human-understandable. However, due to differences in their computation methods, the metrics are not standardized for arithmetic comparison and numerical composition with each other. Nevertheless, there is one exception where M29 on data trust is a weighted sum of data accuracy (M5) and data completeness (M13). Several metrics (M1, M17, M24, M30) use statistical properties of batch data instead of statistical properties of reference high-quality data. For instance, Sanyal and Zhang [33] compute data uncertainty (M30) as the Shannon entropy of batch data without the need for cordoning reference data.

Data quality metrics are extensively studied in the literature but are not widely used by industrial IoT systems having *dark data* [67]. Dark data is unstructured, untagged, and untapped data that has not yet been analyzed or processed. IoT systems accumulate it for various reasons, including compliance and security obligations. Beneficial data becomes outdated because of the lack of tools and processes to use data in a timely manner. Dark data often represents lost opportunities (e.g., revenue, products) for a business as data content and quality is unknown. Almost 90% of IoT data is dark data [83]. If computed promptly and presented as feedback to humans, data quality metrics can instill a company culture to use data before it becomes dark. Furthermore, they can improve data audibility and boost the acquisition of higher-quality data suitable for products based on ML/AI.

RQ2.1 Conclusion. We identified a large spectrum of data quality metrics in the literature. However, we could not find any study reporting the adoption of these metrics in industrial IoT systems as a common practice. We need research to facilitate using quality metrics in industrial IoT settings while addressing the problem of dark data. In the future, it would be interesting to study the perception of human users when AI-driven decisions made on data are presented alongside the data quality metrics.

4.2.2 *RQ2.2: What are the data repair techniques?*

As described in Section 3, data repair techniques restore data lost, accidentally deleted, corrupted, or made inaccessible, while data cleaning techniques only remove corrupt or noisy data. We identified ten primary studies that provide or employ a data repair technique. We excluded studies that do not give any detail, e.g., Wei et al. [45] proposing data interpolation without explaining how to apply it in the CPS/IoT context. Table 5 presents the data repair techniques in the primary studies. The first two columns provide the data quality issue and the data repair technique before a brief description in the third column. The fourth and fifth columns indicate if the repair technique is online (data repair at the data source in real-time) and evaluated. We classify it offline (data repair for large historical datasets on the cloud) if the study does not report any deployment for online data repair.

Missing values are the most data quality issue addressed by the data repair techniques (five primary studies). These techniques use data imputation methods (i.e., replacing missing values with estimates and analyzing the data set as if the imputed values were actual observed values) from statistics to repair missing values. Different studies employ different imputation techniques (median-based [18], mean-based [46], average-value [19], and matrix factorization [44]) for missing value repair. Corrales et al. [9] provide a guided process with data quality techniques (data repair and cleaning). They assume that missing values are represented by special characters such as ?, *, blank spaces, or special words (NaN, null). The user selects one of the imputation techniques offered (hot deck, imputation based on missing attributes, and imputation based on non-missing ones). Most of the repair

Table 5. Data repair techniques in the primary studies.

DQ Issue	Technique	Description of the Technique	Online	Evaluated	Reference
Missing Values	Data Imputation	Missing values are calculated based on the average value of the previous and next data. When there is a wide range of missing data, missing values are filled in according to similar machining data in the manufacturing workshop.	No	Yes	Kong et al. [19]
Missing Values	Median-based Data Imputation	The median value is determined by arranging data in increasing order. Missing values are filled in by the median value. This method underestimates the variance in the dataset since the same median value is used for multiple missing values.	No	No	Khan et al. [18]
Missing Values	Guided Process for Data Repair in Regression models (DC-RM) - Data Imputation	DC-RM provides a procedure for building data repair/cleaning process for regression models. For each data quality issue in the data sets, a data quality task is suggested. In DC-RM, three data imputation techniques are employed: hot deck (missing items are replaced by using values from the same dataset), imputation based on missing values (assigns a value to a missing one based on measures of central tendency), imputation based on non-missing attributes (a regression/classification model is performed).	No	Yes	Corrales et al. [9]
Missing Values	Sensor-network-regularization-based Matrix Factorization - Sensor Substitution - Data Imputation	A correction engine (CE) uses a sensor-network-regularization-based matrix factorization method (SnrMF) to predict missing values. The SnrMF method takes advantage of the correlations among diverse sensors positively correlated. It is capable of improving the performance of reconstructing missing data for a single time series. Moreover, similarity functions are used to determine sensor correlations.	No	Yes	Wang et al. [44]
Missing Values	Mean-based Data Imputation	The approach repairs data of wafer (a collection of microprocessors) production. Some wafers are temporarily split from their parent lots into child lots (a group of wafers processed together in the production). The child lots may undergo single or multiple processes at different times. Their means are used to estimate each wafer's missing values based on lot membership at each process.	No	No	Weiss et al. [46]
Data Veracity	Sensor Calibration using Machine Learning	ML methods can be used for calibrating low-cost sensors; adjusting measurements to compare to concentrations from reference monitors. The approach determines the factors affecting data quality for the given measurements, models their effects on sensor's response, and applies the model to correct the response.	Yes	Yes	Okafor et al. [27]
Data Veracity	Sensory Substitution - Fog based Analytics	The approach validates the initial sensed data with the additional sensed data to reduce false positives and negatives. It focuses on fog-based analytic algorithms using (a) the initial sensed data using cameras-specifically the processed output from the edge, and (b) raw data from another ambient sensor.	Yes	Yes	Russel et al. [31]
Data Veracity with Noise, Outliers, and Missing Values	Data Aggregation Schema based on Shannon's Entropy	A data aggregation scheme for highly uncertain raw IoT sensor data reconstructs the subspace using sample data and then tracks down the low-rank approximation of the dominant space in the presence of high uncertainties at the fog server. The robust dominant subspace is used to estimate a more reliable true sensor data matrix from the highly uncertain raw IoT sensor data matrix.	Yes	Yes	Sanyal and Zhang [33]
Data Veracity	Dependent-Computation Replay Technique	This approach "repairs" corrupted data from its origin through its computational dependencies in a distributed IoT setting. It tracks causal data dependencies and replays dependent computations in event-driven IoT deployment frameworks. It uses histories of persistent storage updates and their causal relationships to update corrupted or approximated data structures with corrected values at the historical point in an application state update sequence at which they occurred. It replays all dependent computation from that point forward.	Yes	Yes	Lin et al. [23]
Outlier (Erroneous values)	Clustering and Regression Modeling	It is high-dimensional and prediction based outlier detection, using a generalisation of (full dimensional) clustering and (full-data) regression modelling. First, the data set is clustered (using K-means, Within Cluster Sum of Square, and Total Sum of Square) to identify the relationships among the data set attributes. Random Sample Consensus (RANSAC) is applied together with the Median Absolute Deviation (MAD) to identify the outliers in the clusters. The new values for the outliers are determined using the overflow, overweight, substitution value, and algebraic sign calculations.	No	Yes	Flick et al. [12]

techniques use data from the same sensor to predict missing values. However, it is not convenient to use the same dataset when there is a wide range of missing values. Kong et al. [19] discuss using data from similar sensors but do not provide any implementation. Wang et al. [44] use correlations among sensors to reconstruct missing values in a single time series. They provide only offline repair.

As mentioned in RQ1.1, data veracity is the accuracy or truthfulness of a dataset and is sometimes considered a data security property. Four studies [23, 27, 31, 33] provide techniques that repair "corrupt" data and increase its accuracy. Three of them [27, 31, 33] use reference sensors/monitors or sample data to improve sensor data accuracy. For instance, Russel et al. [31] present an approach that repairs the initial sensed data from cameras (the processed output from the edge) with the raw data from an ambient sensor. It uses sensory substitution to increase the data robustness, resilience, and dependability. Lin et al. [23] are different from these three studies and repair corrupted data from its origin through computational dependencies in a distributed IoT setting. They replay all the dependent computations to correct data degraded throughout its life cycle due to hardware malfunction, software bugs, or network partitions.

ML has the potential for online and offline data repair in CPS/IoT as correlations among various data sources (sensors) can be learned to substitute one sensor with another sensor and predict missing values or new data that replace corrupt data. Non-AI techniques have different constraints limiting their applicability. For instance, Lin et al. [23] require all dependent data computations in the application state history, which are not always available. ML can support more generic repair solutions for IoT systems having multiple sensors that can replace each other. However, we revealed only two studies [12, 27] applying ML to data repair. Flick et al. [12] use ML (K-means for clustering and regression modeling) only to detect outliers in the clusters, not to predict data replacing outliers. They employ the overflow, overweight, substitution value, and algebraic sign calculations to calculate replacing data. Okafor et al. [27] use linear regression and neural networks to correct sensor output. Their approach determines the factors affecting data quality, models their effects on the sensor response, and applies the calibration model to calibrate sensors. It merges data from multiple sensors into the calibration equation to ensure consistent and accurate information for the model.

The full potential of ML for data repair still needs to be explored with the challenges of integrating ML models and components related to the data and model evolution. Manufacturing is the dominant application domain of data quality research for CPS and IoT (see RQ1.2). Although manufacturing processes produce the same products or parts repetitively, there might be occasional, minor modifications to product specifications and process parameters (e.g., the need to ramp up production) that can render ML models obsolete. Therefore, we need solutions that investigate continual learning [114] and domain adaptation [62] in conjunction with the continuous deployment of ML models [133].

One challenge for data repair research is to create real-time (online) data repair services (e.g., quality monitors and repair services on edge gateway) for IoT systems. However, we identified only four primary studies [23, 27, 31, 33] addressing online data repair. The techniques proposed by these studies detect and repair "corrupt" data at the edge/fog devices close to the data source where computation resources are limited. Only one technique [27] is an ML-based repair solution. It does not address different deployment and versioning scenarios of ML models on edge and cloud. An ML-based data repair technique should be invoked either on edge or cloud to create ML models based on the availability of training data. The models should be containerized as online repair services and deployed on edge for real-time data repair or on the cloud for offline repair.

Only two data repair techniques [18, 46] in our SLR have not been evaluated. They are part of data analytics frameworks. Therefore, the focus of the evaluation in their studies is the data analytics frameworks, not the outcome of the data repair techniques. One study [31] reports the experience with its repair technique and does not quantitatively assess its performance. We investigate the details of the evaluation of the data repair techniques in RQ2.4.

RQ2.2 Conclusion. Across primary studies, data repair techniques address missing values, data veracity, and outliers. Most of these techniques are non-AI solutions having limitations in the industrial CPS/IoT context. ML can support more generic (online and offline repair) repair solutions for IoT systems having multiple sensors that can replace each other. Future research should explore the full potential of ML for data repair and address the challenges of integrating ML models and components related to the data and model evolution.

4.2.3 *RQ2.3: What are the data cleaning techniques?*

Data cleaning techniques detect and remove corrupt and unusable data, e.g., those affected by environmental noise or extreme operating conditions. They do not attempt, for instance, to restore any data deleted or corrupted. Table 6 presents data cleaning techniques in the primary studies. The table structure is similar to the one in Table 5. Table 6 does not include studies that do not give any detail (e.g., Saranya and Sivakumar [34]).

Table 6. Data cleaning techniques in the primary studies.

DQ Issue	Technique	Description of the Technique	Online	Evaluated	Reference
Outlier	DBSCAN-based Outlier Detection	It is a hybrid prediction model that consists of Density-Based Spatial Clustering of Applications with Noise (DBSCAN)-based outlier detection and Random Forest classification. DBSCAN [81] was used to separate outliers from normal sensor data, while Random Forest was utilized to predict faults.	Yes	Yes	Syafrudin et al. [40]
Outlier	Clustering Algorithms	Distance measure and clustering algorithms detect and remove outliers.	No	No	Wei et al. [45]
Noise	Smoothing Filter	The use of smoothing filters [140] is proposed for denoising manufacturing data. The details of how a smoothing filter can be applied are not explained.	No	No	Wei et al. [45]
Outlier	Domain Knowledge-based Heuristic	Some production cycles in the data are not actual production measures but test cycles. The cycle length deciles are used to remove those in the first or last decile.	Yes	No	Cerquitelli et al. [8], Proto et al. [29]
High Dimensionality	Guided Process using a Machine Learning-based Model	A reduced representation of time series is obtained by dimensionality reduction techniques (e.g., Chebyshev polynomial approximation, polynomial regression). An ML-based model combines reduction techniques with extracted features from time series to recommend the most suitable reduction technique.	No	Yes	Villalobos et al. [42, 43]
Noise	Guided Process	An engineer determines if input time series must be denoised. A technique (Frequential Filtering [120] or Moving Average Filter [126]) is automatically proposed based on the input data properties (The automation is not explained).	No	No	Villalobos et al. [42, 43]
Outlier	Domain Knowledge-based Heuristics	Domain specific rules are defined. For example, if the machine speed is lower than 8000, all data are discarded. The machine is supposed to be shut down at 8000.	Yes	No	Yu et al. [49]
Missing Values	Domain Knowledge-based Heuristic	Trajectory data from an agricultural monitoring system is cleaned. The track data of machinery are recorded in equal time. When time difference between adjacent track points is greater than the time interval of isochronal recording, data loss is detected. The lost data location is written into an abnormal data table.	No	No	Hui et al. [14]
Duplicated Records	Domain Knowledge-based Heuristic	If the data recording time of two adjacent track points is the same, these adjacent track points are duplicated records. Data in one of the points is deleted.	No	No	Hui et al. [14]
Data Drift	Domain Knowledge-based Heuristic	Data drift is detected when the position coordinates of the track point deviate from the real position due to the problem of receiving signals from the location equipment. The data record is deleted and written into an abnormal data table.	No	No	Hui et al. [14]
Noise	Noise Filters	Two noise filters are employed to remove noise and preserve anomalies. They are defined as the distance between two moving window units (computation units). The first noise type happens in a short time, and the second one lasts for a while.	Yes	No	Yu et al. [48]
Noise	Filtering using Decision Trees and Random Forest Classification	Sensor data filtering is perceived as a binary classification problem. Multiple decision trees are combined in a random forest. Decision trees are trained on unnormalized data to filter out unwanted events based on raw data.	Yes	Yes	Buschjager and Morik [5]
Outlier	Decision Tree Algorithm	A decision tree algorithm is used to deal with unusual values (outliers). Data outside of minimum and maximum range is discarded. How decision trees are used for data cleaning is not explained in detail.	No	Yes	Saybani et al. [35]
Outlier	Guided Process for Data Cleaning in Regression Models (DC-RM)	DC-RM provides a procedure for data cleaning in regression models. A data cleaning task is suggested for each data quality issues found in the datasets. Candidate outliers are identified and removed through approaches based on Clustering (e.g., DBSCAN [81]) or Distance (e.g., LOF: Local Outlier Factor [66]).	No	Yes	Corrales et al. [9]
Duplicated Records	Guided Process for Data Cleaning in Regression Models (DC-RM)	DC-RM uses the Standard Duplicate Elimination algorithm [65] to detect duplicate records. The records are removed by performing an external merge-sort and scanning the sorted data set.	No	Yes	Corrales et al. [9]
High Dimensionality	Guided Process for Data Cleaning in Regression Models (DC-RM)	DC-RM supports dimensionality reduction with four approaches: filter (features selected based on discriminating criteria), wrapper (features maintained or discarded based on error measures), embedded (features selected when the regression model is trained), and projection (projection of the original space to space with orthogonal dimensions - principal component analysis [53]).	No	Yes	Corrales et al. [9]
High Dimensionality	Principal Component Analysis	Attribute reduction is applied where there are too many unrelated attributes (kind of data) for an application. The principal component analysis [53] is proposed to be used to filter out irrelevant attributes.	No	Yes	Kong et al. [19]
Noise	Sample Reduction	For the state prediction of machine tools, there is a small proportion of nearly-failure state during machine tools' operation. Data imbalance affects the results of a classification algorithm. Part of the running state data is sampled to obtain a smaller set, whose data amount is comparable to that of the nearly-failure data.	No	Yes	Kong et al. [19]

Outlier is the most data quality issue addressed by the data cleaning techniques (seven primary studies). These techniques differ in detecting outliers, while the cleaning task is standard (i.e., removing the detected outliers from the dataset). They use clustering algorithms [9, 40, 45], domain knowledge [8, 29, 49], decision trees [35], or distance metrics [45] to detect outliers. Syafrudin et al. [40] and Corrales et al. [9] use Density-Based Spatial Clustering of Applications with Noise (DBSCAN)-based outlier detection [81] to separate outliers from normal sensor data. Specific rules detecting outliers are defined based on domain knowledge (manufacturing domain). For example, Yu et al. [49] discard all the data points for the machine speed value lower than 8000 since the

machine is supposed to be shut down at that speed. Cerquitelli et al. [8] use the cycle length deciles to detect test cycles of the manufacturing machines and remove the data of these test cycles from the dataset.

Five primary studies [5, 19, 42, 45, 48] provide data cleaning techniques that address data noise. Four of them [5, 42, 45, 48] use filtering to clean noise. For instance, Yu et al. [48] apply noise filters (i.e., the distance between two computation units) to remove two types of noise (the long and short duration of noise). Kong et al. [19] employ sampling to remove noise in the datasets used for the state prediction of machine tools. Data imbalance (a small proportion of nearly-failure state during machine tool operation) affects the prediction results (classification). Part of the running state data is sampled to constitute a smaller data set, whose data amount is comparable to that of the nearly-failure data.

As we noted in RQ1.1, high-dimensionality is crucial, especially for ML applications at the edge, but one of the least addressed data quality issues in our review. Only three data cleaning techniques [9, 19, 42] address high dimensionality. Two techniques [9, 19] employ the principal component analysis [53] to filter out irrelevant attributes. Villalobos et al. [42, 43] provide a guided process using an ML-based model that combines reduction techniques with extracted features from time series to recommend the most suitable reduction technique. The three techniques are offline, and none of them provide online support for reducing high dimensions in input data sets for real-time ML applications.

All the data cleaning techniques employing clustering algorithms [9, 40, 45], decision trees [35], distance metrics [45], noise filters [5, 42, 45, 48], sampling [19], or the principal component analysis [9, 19] are domain agnostic. They may not always detect and clean domain-specific data quality issues (e.g., removing data of machine test cycles [8], data of the machine having a speed value lower than 8000 [49], or trajectory data from an agricultural monitoring system when the data recording time of two adjacent track points is the same [14]). Therefore, we need guided processes using both domain-agnostic data quality monitoring and domain knowledge-based heuristics to detect data quality issues. The existing ones [9, 42, 43] employ only domain-agnostic techniques. Corrales et al. [9] propose the Guided Process for Data Cleaning in Regression Models (DC-RM) that suggests a data cleaning task (e.g., removing outliers) for data quality issues found through a domain-agnostic monitoring technique (e.g., DBSCAN [81]).

We identified five online cleaning techniques [5, 8, 29, 40, 48, 49]. As indicated in RQ2.2, ML has the potential for developing online solutions (and offline solutions too). Two online data cleaning techniques use ML solutions (DBSCAN - an unsupervised learning method utilized in ML algorithms [40] and decision trees and random forest classification [5]). They do not address the ML model deployment and versioning challenges for online data cleaning. Their primary studies do not report on model deployment and versioning scenarios on edge and cloud (see RQ3.1).

Half of the data cleaning techniques [8, 14, 29, 42, 43, 45, 49] in our SLR have not been evaluated. They are part of predictive maintenance or smart manufacturing [8, 29, 45, 49] and agriculture machinery monitoring systems [14]. Therefore, the focus of the evaluation in the primary studies is not the outcome of the data cleaning techniques. We investigate the evaluation details of the data cleaning techniques in RQ2.4.

RQ2.3 Conclusion. Existing cleaning techniques address outliers, noise, high-dimensionality, duplicated records, data drift, and missing values. Most of these techniques are domain agnostic and may not always be able to detect and clean domain-specific data quality issues. Further research is needed to propose guided processes using both domain-agnostic data quality monitoring and domain-knowledge-based heuristics. There is also a need for online data cleaning support to reduce high dimensions in input data sets for real-time ML applications.

4.2.4 *RQ2.4: How are data quality techniques evaluated?*

Table 7. Metrics used to evaluate data quality techniques.

Evaluation Metrics	Context	Generic	Evaluation Target	References
MAE, RMSE, R^2	Repair	Yes	Data repair approach (regression models)	Okafor et al. [27]
Accuracy	Repair	Yes	Classification Models for Tool Wear Prediction	Kong et al. [19]
RMSE	Repair	Yes	Data repair approach	Wang et al. [44]
Sensor Data Estimation Error	Repair	No	Data repair approach	Sanyal and Zhang [33]
MAE, MSE, RMSE, R^2	Cleaning	Yes	Regression Models for RUL Prediction	Saranya and Sivakumar [34]
MAE	Cleaning	Yes	Regression Models	Corrales et al. [9]
Precision, Recall, Accuracy	Cleaning	Yes	Classification Models for Fault Prediction	Syafrudin et al. [40]

Table 9. Precision, recall, and accuracy metrics for classification models.

Metric	Description	Formula
Precision	The ratio of true positive to the total predicted positive	TP/(TP + FP)
Recall	The ratio of true positive to the total actual positive	TP/(TP + FN)
Accuracy	The ratio of correct predictions to total observations	(TP + TN)/(TP + TN + FP + FN)

In this section, we discuss the evaluation metrics used and reported in the selected papers, their calculation methods, and their strengths and drawbacks.

Eight metrics have been used to assess data quality techniques. They can be categorized as *classification* and *regression* metrics. The former includes precision [40], recall [40], and accuracy [19, 40]. The latter contains Mean Absolute Error (MAE) [9, 27, 34], Mean Squared Error (MSE) [34], Root Mean Squared Error (RMSE) [27, 34, 44], coefficient of determination (R^2) [27, 34], and the true sensor data estimation error [33].

Table 8. Confusion matrix of a classifier.

	Classified as "Yes"	Classified as "No"
Actual "Yes"	True Positive (TP)	False Negative (FN)
Actual "No"	False Positive (FP)	True Negative (TN)

Table 7 lists the metrics used to evaluate data quality techniques. These metrics have been mainly used to evaluate the data quality techniques by using predictive analytics output (e.g., fault and remaining useful life prediction). The main goal is to assess the impact of the data quality techniques on the performance of the classification and regression models. For instance, Kong et al. [19] use the accuracy metric to show that the data processed by the proposed data quality technique improves the classification accuracy for tool wear prediction.

Classification Metrics. As seen in Table 8, classification models have four possible outcomes. True positive (TP) and true negative (TN) denote the correctly classified points. False positive (FP) represents the points incorrectly classified as “yes” (positive) when they are actually “no” (negative). And false negative (FN) refers to the points incorrectly classified as “no” (negative) when they are actually “yes” (positive). According to the definitions of TP, TN, FP, and FN, Table 9 presents the precision, recall, and accuracy metrics.

Syafrudin et al. [40] calculate the precision, recall, and accuracy of classification models predicting faults with and without removing outliers (data cleaning). Although we can use precision and recall to assess the performance of data cleaning solutions (e.g., the number of correctly removed outliers over all the data points removed), we did not find any study using these metrics for that purpose. The accuracy metric uses all TP, TN, FP, and FN in Table 8 and is adequate for only balanced data sets. IoT and CPS data sets obtained from sensors are imbalanced; they usually have more normal data points than erroneous ones, and the class distribution is not even in these data sets. Therefore, the accuracy metric is unfair while assessing data quality techniques on sensor

data sets. It might be why the primary studies ([19, 40]) use the accuracy metric only to evaluate the impact of data quality techniques on the performance of classification models.

Regression Metrics. The MAE, MSE, RMSE, and R^2 metrics have been mostly used to assess the performance of the data quality techniques on the regression models (see Table 7). MSE is the average squared difference (error) between the predicted and observed values. It gives more weight to big differences. It might underestimate the model's accuracy as one big difference might increase the MSE significantly. RMSE is the square root of MSE and is more interpretable. MAE is the average of the absolute differences. Unlike MSE or RMSE, it is less sensitive to big differences since it does not take the square of the errors. R-squared (R^2) represents the proportion of the variance for a dependent variable explained by an independent variable(s) in a regression model. It can be more informative than MAE, MSE, and RMSE, as it can be described as a percentage, whereas MAE, MSE, and RMSE have arbitrary ranges.

Corrales et al. [9] compare the MAE of the regression models trained with the data set not cleaned versus those trained with the same data set cleaned by their data cleaning approach. They show that the results achieved by the trained models with the cleaned data set are better than or equal to that with the same data set not cleaned. Saranya and Sivakumar [34] use the MAE, MSE, RMSE, and R^2 to assess the impact of the proposed data cleaning technique on the prediction of Remaining Useful Life (RUL). They calculate and compare the metrics for the RUL prediction with and without outliers. Different from these two works mentioned above, Okafor et al. [27] employ the MAE, RMSE, and R^2 metrics to assess the performance of their proposed data repair technique. They compare the sensor measurements before and after data repair to reference measurements using these three metrics. Wang et al. [44] use the RMSE to evaluate the performance of missing value prediction. They compare the RMSE of their proposed data repair approach and of the competing methods (e.g., non-negative matrix factorization [106] and support vector machine [139]). Sanyal and Zhang [33] propose a specialized metric for sensor data estimation error to compare their data repair approach with Principal component analysis (PCA) [53], i.e., a classical tool for low dimensional linear subspace approximation, as a baseline algorithm in the presence of high Gaussian noise with outliers and missing values. This metric is similar to RMSE as it is the square root of the sum of the squares of the coordinates of the vectors of the estimated and observed sensor data from each IoT node.

As mentioned above, only three studies (i.e., [27, 33, 44]) use regression metrics to assess the performance of data quality techniques, not their impact on the performance of classification/regression models. They evaluate data repair approaches predicting missing or corrupted data. Therefore, they use regression metrics, i.e., MAE, RMSE, and R^2, that quantify the difference in the estimated and observed values. Although not found in the primary studies, precision and recall can assess the performance of data cleaning techniques. We could not identify a single study evaluating both the performance of a data quality technique and its impact on the performance of a data analytics solution.

RQ2.4 Conclusion. Across primary studies, the metrics are standard and have been mostly used to assess the impact of the data quality techniques on the performance of predictive analytics (e.g., fault and remaining useful life prediction). Few studies assess the performance of data quality techniques. No study evaluates both the performance of a data quality technique and its impact on the performance of predictive analytics.

4.3 RQ3: What software engineering solutions are used for data quality for CPS and IoT in Industry 4.0?

Software is the heart and soul of any IoT system and CPS, especially for analyzing data and its quality. Software engineering solutions for CPS and IoT systems often span the edge-fog-cloud continuum; various software design choices are made based on the requirements for real-time and historical data processed by these systems. To better understand the role of software engineering in data quality, this research question investigates software

engineering solutions in the primary studies. We use the term software engineering solution to cover any software engineering technique (including data storage technologies, programming languages, libraries, and platforms) used to implement data quality techniques for CPS and IoT. To respond to RQ3, we address the following three subquestions:

4.3.1 ***RQ3.1: What programming languages and solutions are used to manage data quality?***

Nineteen papers (out of 51 papers considered in our survey) mention a programming language or solution (e.g., programming platforms, libraries, and models). Table 10 presents the programming languages and solutions used to manage data quality in the papers. Nine of these papers (i.e., [2, 15, 17, 29, 34, 40, 49–51]) propose a data analytic solution as part of a CPS or IoT system. Their main goal is not to provide a data quality technique. They support data quality as a pre-processing step of their data analytics solution. Since these solutions attempt to quantify uncertainty and reason with incomplete and inconsistent data, more *right* data generally results in a better output of such solutions [88]. Python, Java, and R are used to implement data analytics in the studies. ML libraries TensorFlow [52], Weka [91], MLLib [118], scikit-learn [129], and Keras (high-level API of TensorFlow 2) [89] provide a proper abstraction to facilitate the development of the proposed data analytics solutions.

Only ten papers mentioning programming languages and solutions (i.e., [4, 5, 9, 10, 21, 23, 26, 27, 35, 39]) address data quality techniques for CPS and IoT applications as their primary goal. For instance, Lin et al. [23] propose a new approach for repairing corrupted data in IoT applications. It automatically tracks causal data dependencies and replays dependent computations across multi-tiered IoT deployments. It combines the function-as-a-service (FaaS) programming model with versioned, persistent storage and causal event tracking to facilitate data repair. The approach extends an open-source, distributed, FaaS runtime system called CSPOT [151]. CSPOT runs over various devices (e.g., microcontrollers, edge, and public clouds) and makes data repair possible in a distributed IoT setting.

Semantic technologies have been investigated to enable the integration and interoperability of data produced by heterogeneous IoT devices. Bambgboye et al. [4] present a layered software framework using semantic technologies to maintain the consistency of data streams produced by physical sensors. The framework applies semantic modeling and reasoning to validate data stream consistency while highlighting the temporal characteristics of the stream. The framework has four layers: the *sensing*, *modeling*, *reasoning*, and *application* layers. The sensing layer receives data from sensors and prepares the data for the upper layer. It contains a stream service module that ensures the continuous transfer of data streams with Java infrastructure Apache Camel [68]. The modeling layer provides an ontology to integrate and enhance reasoning for sensor streaming data available as raw numeric data. Semantic reasoning achieves the continuous validation of the sensor stream. The reasoning layer validates sensor readings within a time window against prevailing disturbances with data validation policies and other related sensor readings. To this end, the framework layers the Jena rule language [69] with the C-SPARQL query engine [64] for continuous queries over RDF data streams.

The data collected, processed, and exchanged at each stage of CPS and IoT applications might have a different structure, format, and velocity and be stored in data silos not available to all the system users but only to some users. Cui et al. [10] propose a systematic approach, i.e., Data Control Module (DCM), that uses state-of-art big data software to manage data silos in manufacturing. A data silo concerns timely monitoring of data changes, redundancy, inconsistency, and insecurity. DCM employs big data software (Apache NiFi [125], Apache Phoenix [131], and Apache Kafka [99]) to implement functions addressing these concerns. The DCM architecture consists of DCM Cloud and several DCM Edge systems. Apache NiFi takes responsibility for data monitoring at the DCM edge and data collection and allocation at the DCM cloud. It provides a data provenance function to trace data history and check data consistency. Apache Kafka is a high-throughput, low-latency messaging framework. Therefore, it transmits control messages (including data information such as source location, data source computer, expected location, and target data computer) between the DCM Cloud and Edge.

Table 10. Summary of the programming languages and solutions used to manage data quality.

		Languages			Machine Learning Libraries				Bigdata Platforms				Other			
		Java	C/ C++	Python	R	Tensor-Flow [52]	Weka [91]	MLLib [118]	scikit [129]	Keras [89]	Nifi [125]	Camel [68]	Kafka [99]	Spark [142]	Jena [69]	C-SPARQL [64]	Matlab [116]	CSPOT [151]	NOS [132]
Data Quality as a Primary Goal	Lin et al. [23]																	✓	
	Saybani et al [35]																✓		
	Buschjager et al. [5]		✓	✓					✓										
	Corrales et al. [9]	✓			✓														
	Cui et al. [10]										✓		✓						
	Okafor et al. [27]			✓		✓				✓									
	Bamgboye et al. [4]	✓										✓			✓	✓			
	Mohamed et al. [26]			✓									✓						
	Kuemper et al. [21]									✓									
	Sicari et al. [39]																		✓
Data Quality as a Secondary Goal	Zellinger et al. [51]					✓								✓					
	Zacarias et al. [50]		✓	✓	✓		✓												
	Kufner et al. [17]			✓		✓			✓										
	Yu et al [49]													✓					
	Apiletti et al. [2]												✓	✓					
	Proto et al. [29]							✓						✓					
	Saranya et al. [34]		✓						✓										
	Huru et al. [15]												✓	✓					
	Syafrudin et al. [40]												✓						

RQ3.1 Conclusion. The programming languages and solutions used to manage data quality (e.g., TensorFlow, Keras, MATLAB's fuzzy logic toolbox, Jena, C-SPARQL) highly depend on the solution domain (e.g., ML, data mining, semantic web). For instance, Python is almost the de-facto programming language in the studies providing ML-based solutions. Not many papers mention a programming language, but Python, Java, C++, and R are used to implement data quality techniques in the approaches we studied in our survey. Big data software platforms such as Apache Kafka and Phoenix are suitable for implementing data quality concerns for high-velocity data, such as the timely monitoring of data changes, data redundancy, data inconsistency, and data insecurity.

4.3.2 *RQ3.2: What data storage solutions are used to manage data quality?*

Data storage support depends on data type, where data is stored (Local/Edge/Cloud), and in which context data is processed. We analyze the relationship between data storage support (e.g., time-series database systems, NoSQL, or a blockchain solution) and data quality. We expect to gain new insights into essential data storage solutions in CPS, IoT, and Industry 4.0 applications and their impact on data quality techniques such as data cleansing and repair. These new insights will help organizations choose appropriate data storage for data quality.

Twenty papers (i.e., [1, 2, 7, 10, 11, 14–17, 19, 23, 24, 26, 32, 37, 39, 40, 42, 44, 49]) mention data storage support, e.g., database, file system, or a blockchain solution, where data storage is mostly part of data analytics frameworks. Three papers [1, 7, 11] employ the blockchain to ensure data security, such as the trustworthiness of sensor observations, while the blockchain is also a storage medium. There is an explicit dependency between the data storage solution and the data quality support (i.e., ensuring data security) in these two works. Mohammed et al. [26] employ a cloud-based solution, i.e., the google cloud environment, to store IoT data. Some papers refer to some domain-specific database systems (not any well-known database system) without detailed information, such as agricultural telematics [14] or operation management database [16]. They do not provide any insight into database technologies, how data is managed in what format and quality, and the impact of the database technologies on the data quality techniques.

Data warehouses and distributed file systems are ideal mediums for storing data for big data systems [32, 48, 49] that receive data from multiple data sources. Such data systems require data cleansing before data gathered from various sources are integrated. They may use data warehouses and distributed file systems combined with databases in a layered fashion where each layer has its data cleansing. For instance, Santos et al. [32] propose data storage layers having different components used in various contexts: (a) data streams are stored in a real-time fashion in a NoSQL database in the real-time data storage layer; (b) the Staging Area and Big Data Warehouse components save data in a more historical perspective in the historical data storage layer. In the Staging Area component, the Hadoop distributed file system stores data that are available for further use for a limited time. Shah et al. [37] propose a plug-and-play solution to use the data storage layers as an interface to data storage. This decoupling enables the data storage technologies (e.g., data warehouses, relational databases, and distributed file systems) to be easily replaced based on the type of stored data. However, replacing the data storage component accessed through the layer may require changes in the data quality techniques (e.g., data cleansing) due to the data quality support the new component provides.

The papers do not report a direct relationship between the database technologies and the data quality techniques. Table 11 gives a classification of database systems and the solutions using these systems. We use the database management system taxonomy provided by Gudivada et al. [88]. In addition to the database classes in Table 11, Gudiva et al. mention Native XML, RDF Stores, and Key-Value Stores database classes which none of the papers in our survey report. Most papers report the use of column-family database systems (i.e., HBase and Cassandra) since these systems support heterogeneous data and tolerate network failures and temporal data inconsistency. A

Table 11. A classification of database systems in the primary studies.

DB Class	Used DBs	References	Description of the DB Class by Gudivada et al. [88]
Column Family	HBase	Kong et al. [19], Huru et al. [15], Wang et al. [44], Cui et al. [10]	Ideal for storing sparse, non-transactional, and heterogeneous data and retrieving partial records; accommodate flexible and evolving database schema; tolerance to both network failures and temporary data inconsistency; increased processing power through horizontal scalability.
	Cassandra	Villalobos et al. [42], Apiletti et al. [2]	
Document Oriented	MongoDB	Villalobos et al. [42], Syafrudin et al. [40], Sicari et al. [39]	Ideal for managing semi structured, arbitrarily nested hierarchical document data organised in the form of key-value pairs in JSON format; support flexible schema evolution; accommodate high data variability among data records.
Relational DBMS	sqllite	Kufner et al. [17]	Two subclasses: row- and column-oriented. Row-oriented: optimised reads and writes for online transaction processing; enforces strong data integrity; provides transaction support, data distribution and replication, and fine-grained access control. Column-oriented: optimised reads for online analytical processing; enforces data integrity and provides distributed data analytics.
	Apache Phoenix	Cui et al. [10]	
	a cloud-based relational DB	Liu et al. [24]	
Time-series DBMS	InfluxDB	Villalobos et al. [42]	Ideal for storing and retrieving time series data, which is data indexed by time; efficient execution of range queries; performance at scale; support for age-based data retention and archival.
Graph-Oriented	Neo4J	Villalobos et al. [42]	Ideal for storing and flexible querying relationship-rich data; powerful operators for graph traversals and identifying subgraphs and cliques based on relationship types

column family is a NoSQL database containing columns of related data. The column-family database systems such as HBase and Cassandra also support time-series data.

RQ3.2 Conclusion. We derived the following insights into essential data storage solutions in CPS, IoT, and Industry 4.0 applications and their impact on data quality techniques: (i) blockchain is an ideal solution to ensure data security as part of data quality; (ii) big data systems gathering data from various sources combine multiple data storage solutions, which require a layered data storage architecture where each layer may require its own data quality technique; (iii) there is no direct relation between the key database technologies and the data quality techniques; and (iv) the column-family database systems are highly preferred since they support heterogeneous data and tolerate network failures and temporal data inconsistency.

4.3.3 *RQ3.3: What IoT reference architecture layers are covered in the primary studies?*

Data can be processed/stored at different levels of the IoT reference architecture (see Section 3.3). These architecture levels can help understand how and where data is analyzed and processed for quality issues. We have identified twenty primary studies that refer to the IoT architecture layers (see Table 12 and **L1**-**L7** in Section 3.3).

Five primary studies [1, 14, 26, 42, 44] focus on data quality management in the cloud-based architecture layers (**L4**, **L5**, and **L6**). Three of these studies [14, 42, 44] propose data quality management techniques (data repair and cleaning) running on the cloud. These techniques are offline, e.g., data repair for large historical datasets on the cloud. Sensor data in motion are converted for long-term storage and stored in an easily accessible format on the cloud to be further processed, e.g., for historical data validation. For instance, Villalobos et al. [42] focus on the layers **L4** and **L5** for time series data captured by machine sensors and accessed using a REST API via a gateway. The remaining two primary studies present a cloud-based approach to measure IoT data freshness [26] and a distributed architecture using blockchain and smart contracts for data quality in logistics traceability [1].

Six primary studies [5, 7, 11, 13, 17, 41] focus on data quality management in edge-based architecture levels (**L2** and **L3**). Only one study [5] provides a data quality technique running on edge (**L3**). It is an online data cleaning technique that uses ML (decision trees and random forest classification) to filter out unwanted events on raw data on the edge before further processing data on the cloud. One study [7] proposes a game theory algorithm running on edge (**L3**) to detect fraudulent data. Dedeoglu et al. [11] use gateways (**L2**, **L3**) to calculate trust for sensor observations. Guo et al. [13] distribute portions of large volumes of data from machines (**L1**) to the edge (**L3**). Kufner et al. [17] combine signal acquisition and concurrent analysis techniques in a distributed edge-based structure (**L2**, **L3**) to achieve vertical data continuity.

Nine primary studies [4, 8, 10, 16, 23, 24, 31, 40, 49] cover edge-cloud orchestration for data processing (not necessarily data quality management). Four of these studies [23, 31, 40, 49] provide a data quality management technique (data cleaning and repair) in the architecture layers from **L1** to **L7**. They are all online techniques and implement data monitoring and cleaning/repair across the edge-cloud continuum. For instance, Syafrudin et al. [40] incorporate distributed gateways (**L3**) that obtain sensor data and application layers (**L4**, **L5**, and **L6**) used for detection/removal and fault prediction. The remaining five primary studies mention data quality as part of data analytics support or data management frameworks. Therefore, they cover almost all the IoT architecture layers while addressing data quality management in a subset of these layers.

Table 12. IoT reference architecture layers in the primary studies.

Primary Study	IoT Architecture						
	Perception	Network		Application			
	L1	L2	L3	L4	L5	L6	L7
Villalobos et al. [42]	✓			✓	✓		
Wang et al. [44]	✓			✓	✓		
Hui et al. [14]	✓			✓	✓	✓	
Mohammed et al. [26]	✓			✓	✓		
Ahmed et al. [1]	✓			✓	✓		
Tham et al. [41]	✓		✓				
Kufner et al. [17]	✓	✓	✓				
Casado-vara et al. [7]	✓	✓	✓				
Dedeoglu et al. [11]	✓	✓	✓				
Guo et al. [13]	✓		✓				
Buschjager et al. [5]	✓		✓				
Yu et al. [49]	✓		✓	✓	✓	✓	✓
Liu et al. [24]	✓	✓	✓	✓		✓	✓
Cerquitelli et al. [8]	✓	✓	✓	✓	✓	✓	✓
Bamgboye et al. [4]	✓		✓	✓	✓		
Lin et al. [23]	✓		✓	✓	✓		
Syafrudin et al. [40]	✓		✓	✓	✓	✓	
Jeong et al. [16]	✓	✓	✓	✓	✓	✓	
Cui et al. [10]	✓		✓	✓	✓	✓	
Sanyal et al. [33]	✓	✓	✓	✓	✓		
Total (✓):	20	7	16	14	13	7	3

✓ = the contribution specifies the IoT architectural aspects from the taxonomy

Our observation is that the ubiquitousness, complexity, and size of IoT systems pose fundamental challenges and limitations in deploying data quality techniques across the IoT reference architecture layers. IoT systems may run on several IoT device types (e.g., smart camera, thermostat, smart tv, force sensors, vibration sensors) having different operational environments (e.g., industrial, enterprise, consumer). These devices may operate on several protocols (MQTT, CoAP, AMQP) and connection types (device-to-device, device-to-gateway, gateway-to-data systems) with different data acquisition systems. Having several device and protocol configurations for IoT systems obtaining different kinds of data in different operational environments leads to several resource constraints and data quality management scenarios (online and offline). For instance, we deploy a data repair technique on the cloud for the historical (offline) repair of high-frequency manufacturing data stored in the cloud infrastructure. The same technique may also be deployed on edge to perform in-motion (online) data repair for real-time predictive maintenance. We need highly-configurable data management techniques deployed on different layers of the IoT reference architecture for different scenarios, e.g., on a standalone machine, edge device, or the cloud, with access to a long or short-term database or an API provided by a data acquisition system. However, the current data repair and cleaning techniques we summarized above address particular scenarios (online or offline). They are not deployed on different architecture layers in the edge-cloud continuum for different quality management scenarios based on the needs of the targeted IoT system.

As we mentioned in RQ2.2 and 2.3, data quality monitoring is part of data cleaning and repair. However, there might be cases where the data repair or cleaning technique should run together with several data quality monitoring techniques deployed on different architecture layers. For instance, the same repair technique may

predict values for missing values identified by a data monitoring approach on edge and values replacing erroneous data identified by another monitoring approach in the cloud as part of historical data validation. Applying decentralized ML architectures to data quality management might address all these challenges and limitations of the existing techniques related to their deployment on the architecture layers depending on the quality management requirements and scenarios. We can distribute the ML model training for data repair and cleaning and containerize the trained models to be deployed on a standalone machine, edge, or the cloud for different quality management scenarios. On the other hand, we need research to address the data and model evolution challenges and limitations caused by integrating ML models and components (see Section 4.2.2).

RQ3.3 Conclusion. Most existing research focuses only on data processing in (indirect) relation to data quality without considering other aspects within the IoT architecture. We found very few studies discussing all the layers of IoT architecture where data flow from IoT devices, via edge, to the cloud. The existing data quality techniques do not cover different combinations of IoT reference architecture layers for different scenarios depending on the needs of the targeted IoT system. We need further research on data quality techniques that can run on a standalone machine, edge device, or the cloud, with data access to support online and offline data repair and cleaning.

4.4 Summary of Data Quality for CPS and IoT in Industry 4.0

Figure 5 summarizes the results related to our RQs. It shows the data quality issues for CPS and IoT, the sources of these quality issues, the data quality metrics, the data quality techniques, the metrics used to evaluate the techniques, and the software engineering solutions to manage data quality. It can be used with the taxonomy of data quality in data-driven paradigms (see Figure 2) to classify future data quality research for CPS and IoT in Industry 4.0.

5 RELATED WORK

Several works study the literature on CPS and IoT. The focus of most of the surveys for CPS is on security and privacy (e.g., [84, 85, 93, 96, 102, 112, 113, 119, 124]). Some of them [84, 96, 102] survey the literature for CPS security from a more general perspective; some others focus on specialized security topics such as intrusion detection systems [119], deep learning-based anomaly detection [113], physics-based attack detection [85], differential privacy [92], and model-based security engineering [124] for CPS. Gunes et al. [90] and Chen et al. [71] conduct secondary studies on the applications and challenges of CPS. Dey et al. [77] focus on the research for CPS in the medical domain. Xu et al. [154] survey the literature on the intersection between CPS and big data in Industry 4.0. Like the secondary studies for CPS, various studies for IoT address security and privacy (e.g., [54, 56, 59, 70, 72, 73, 93, 94, 108, 115, 117, 122, 146]). Some other studies are specialized in interoperability for Industrial IoT (IIoT) [95], IoT protocols, technologies, and applications along with related issues [57, 61, 73, 78, 107, 123, 137], IoT applications in blockchain systems [105], applications of blockchain technologies to IoT [158], IoT big data [63], data analytics for IoT [141], IoT-based smart cities [60], IoT for agriculture [80], IoT for healthcare [101], and IoT in industries [76]. Some secondary studies (e.g., [153]) survey the works for CPS and IoT in the context of Industry 4.0. Some other studies (e.g., [109, 111, 128, 136, 155, 157]) mainly focus on the literature for Industry 4.0, where CPS and IoT are considered building blocks of Industry 4.0.

None of the works mentioned above address data quality for CPS and IoT. We have identified only three SLRs and three surveys focusing on data quality in the context of CPS and IoT, as briefly presented in Table 1. For each related work in the table, the symbol '✓' indicates that the work provides the feature, the symbol '✗' indicates that it does not provide the feature, and 'NA' indicates that the required information is unavailable.

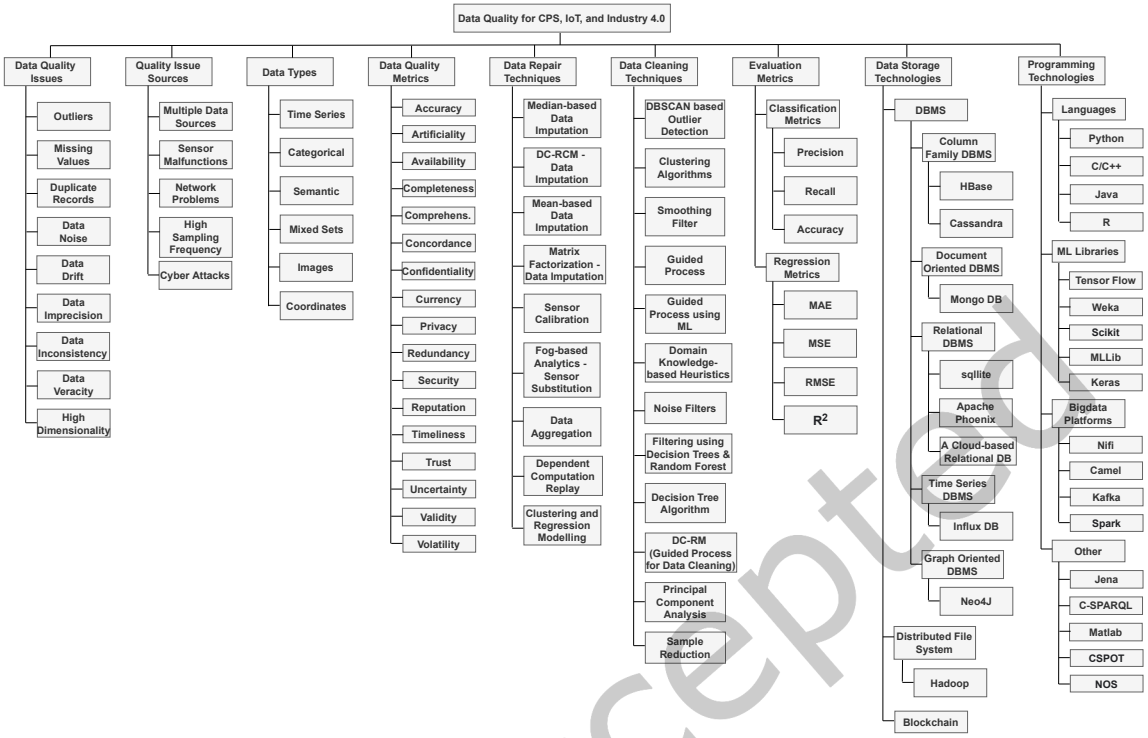

Fig. 5. Summary of data quality in CPS and IoT for Industry 4.0.

Karkouch et al. [100] surveyed fourteen papers published between 2012-2016. Their paper selection was random without a systematic search and selection process of systematic reviews [104, 130, 150]. Moreover, we could not find any information on how the authors chose the discussed aspects of data quality for IoT (DQ classification schema). Karkouch et al. listed four data quality challenges for IoT data: (i) the scalability of cleaning methods in distributed systems, (ii) the heterogeneity of data sources requiring complex approaches, (iii) the automated verification without human interaction, and (iv) the fail-safe distributed architecture. They highlighted the need for an abstraction level that supports data quality assessment independent from data types.

Wang and Wang [148] reviewed the state-of-the-art for time series data cleaning and classified time series errors. They mentioned four challenges related to time series data cleaning: (i) a large amount of data with a high error rate (esp. in industrial settings), (ii) ambiguous reasons for errors, (iii) continuous nature requiring online analysis, (iv) minimum modification principle. They identified the need for research for analyzing error types defined rather broadly. They highlighted the lack of multivariate cleaning algorithms and the potential for utilizing ML for data cleaning.

Zhang et al. [156] compared twenty-one IoT data quality frameworks and five related standards. The diversity in data quality frameworks and various definitions of data quality dimensions and metrics hampered the comparability of the survey. The survey revealed the need for a more user-friendly data quality assessment methodology based on existing generic frameworks. Teh et al. [144] presented a recent SLR on sensor data quality problems. They mainly investigated the error types of sensor data and sensor data error detection and correction methods. Unlike our SLR, their SLR did not study the application domains for data quality research for IoT, the trade-off between data quality and security, and software engineering solutions used to manage data quality. Teh

et al. reported several methods proposed to detect and correct sensor data errors. 90% of the studies in the SLR provided proper validation, and 68% used not publicly available or reproducible data. Teh et al. highlighted that the methods are not comparable without further ado since variations of a common idea are utilized in many cases with a varying research methodology (e.g., labeling, error injection, or preprocessing method). They revealed the need for a benchmark system to compare data quality techniques.

Liu et al. [110] conducted an SLR on data quality in IoT based on 45 empirical studies from 2012 to 2018. Contrary to our study, their SLR is limited to data quality problems (issues), dimensions, and measures. It does not cover topics such as data quality techniques, their evaluation, application domains for data quality research, and software technologies for managing data quality, which are the main points of our SLR. Liu et al. established links among data quality dimensions, manifestations of data quality problems, and methods utilized to measure data quality. They identified the potential areas for future work: (i) developing guidelines for defining specific data quality dimensions of IoT data, (ii) addressing data quality problems based on different IoT layers, and (iii) constructing data quality frameworks in the IoT context.

More recently, Alwan et al. [58] conducted another SLR to investigate data quality challenges in smart cities as large-scale CPSs and identify the most common techniques used to address these challenges. The scope of the SLR is limited to the data quality challenges for CPSs, the data quality techniques to overcome these challenges, and the effectiveness of these techniques. The SLR does not cover data quality metrics, data security, or software engineering solutions to manage data quality. Similar to the results we reported (see Subsection 4.1.1), Alwan et al. revealed that data quality issues occur in large-scale CPSs because of sensor malfunctions, calibration issues, poor sensor node quality, environmental effects, external noise, and networks or communication errors. They categorized the data quality solutions into three primary groups: *data mining*, *technical models*, and *mathematical models*. Data mining methods (i.e., anomaly detection, classification, clustering, and predictive analysis) are the most widely used compared to others.

In comparison with the SLRs and surveys mentioned above, our SLR investigates exclusively three aspects of data quality for IoT, CPS, and Industry 4.0: (i) data quality problems (including data quality issues, their resources, application domains, data types, and data quality and security trade-off), (ii) data quality techniques to overcome the problems (metrics to monitor data quality, approaches for data repair and cleaning, and evaluation of these approaches), and (iii) software engineering solutions for data quality (architectures, programming languages, and data storage solutions). Existing SLRs focused on one or two aspects with a limited scope. For instance, Liu et al. [110] studied the literature on data quality problems based on quality dimensions and measures. We approach data quality research for theoretical and practical implications in a much broader scope. We also report some research directions.

6 THREATS TO VALIDITY

Our systematic literature review addresses a wide range of approaches and domains. Our review process had automated (e.g., search queries) and manual (e.g., data extraction) parts. Therefore, some relevant studies and information might have been uncovered. In the following, we summarize several measures taken to mitigate this issue.

6.1 Internal Validity

Search queries. We aimed to find as many relevant publications as possible by using general terms related to data quality in our search queries. We used our inclusion and exclusion criteria to select the papers. It is still possible that we should have included more research in the final selection of primary studies. To mitigate this risk, we conducted a manual search to limit the possibility of missing studies throughout the database search process. We discovered most of the primary studies through our database search. The search features provided

by these online publication databases are not always the same, which may lead to misleading search results. To mitigate this risk, we adapted our search string for the built-in search features of each database.

Study inclusion and exclusion. Even though we have well-defined inclusion and exclusion criteria, including or excluding a study could still be subjective, especially when its contribution is indirect to data quality (e.g., some studies support data quality as a preprocessing step of data analytics). To mitigate this risk, we conducted cross-checks between at least two authors and then group discussion to remove papers that did not have enough scientific contribution according to our selection criteria.

Data extraction. Missing and misinterpreting information is a risk of manual information extraction. To mitigate this risk, we distributed the primary studies to the authors for data extraction. They later validated the data extraction for each other. One co-author reviewed the data extraction and its validation to identify and summarize the data extraction inconsistencies. We settled all the conflicts during meetings with the authors.

6.2 External Validity

The online repositories we used in our SLR restrict our review results. To mitigate this risk, we employed repositories that well-known venues have been included in and that previous survey papers have extensively used.

6.3 Conclusion Validity

The primary studies have a variety of application domains, and some of them have primary goals different than data quality (e.g., data analytics), which made it difficult to determine direct contributions to data quality and draw decisive conclusions. To mitigate this issue, we categorized the primary studies based on goals (e.g., data quality as a primary/secondary goal in Table 10) for our research questions.

6.4 Reliability Validity

The readers can replicate our systematic literature review study if they follow the steps of our review process. It is still possible to have some inconsistent results because of potential differences in the manual steps of the review process, such as data extraction and synthesis. To mitigate this risk, we provided the details of our review process (see Section 2) and a summary of data quality research (Figure 5) that can be used with our taxonomy of data quality (Figure 2) in data-driven paradigms to classify and compare future primary studies.

7 CONCLUSIONS

This paper presented the results of our systematic literature review (SLR) regarding data quality research for CPS and IoT in Industry 4.0. Obtaining data from IoT and CPS for decision support is crucial to improving efficiency and competitiveness in many industrial sectors and application domains. Data quality techniques ensure the input quality for decision support and become an inherent component of data-centric CPS and IoT applications. We followed the common SLR steps (i.e., the definition of research questions, a search strategy, inclusion and exclusion criteria, and data synthesis and extraction method) to conduct our review.

Our systematic search and selection process yielded 51 primary studies published between 2011 and 2021 (and three SLRs and three surveys). The growing number of studies in recent years indicates an increasing interest in data quality research for CPS and IoT. Our objective was to analyze how data quality has been treated for data-centric CPS and IoT applications and evaluate what the proposed data quality techniques have done for those applications. We investigated data quality issues and their sources for CPS and IoT, data quality metrics, data quality techniques (including data quality monitoring, data repair, and data cleaning), and software engineering solutions for handling data quality. To answer three RQs (ten sub-questions), we obtained and synthesized data from the primary studies. From our SLR, we conclude the following:

(a) The primary studies address a variety of data quality issues. Outliers, missing values, and data veracity are the three main ones. Not many studies discuss the source of data quality issues and the implications of different computing architectures for data quality issues. Future research should further investigate the reasons and the implications of computing architectures for data quality issues to devise better techniques.
(b) We could not find any study reporting the adoption of data quality metrics in industrial systems. We need research to facilitate using quality metrics in industrial settings.
(c) Non-AI data repair solutions have limitations in the industrial CPS/IoT context (e.g., requiring dependent data computations in the application state history, which are not always available). Future research should explore machine learning for online and offline data repair while addressing model deployment and evolution.
(d) Most data cleaning techniques are domain agnostic and may not always detect and clean domain-specific data quality issues. Further research should address guided processes that use domain-agnostic data quality monitoring and domain-knowledge-based heuristics. Real-time machine learning applications need online data cleaning support to reduce high dimensions in input data sets.
(e) Existing data quality management techniques do not support deployment on different IoT layers for online and offline scenarios. Future techniques should be able to run on a standalone machine, edge device, or the cloud, with data access to support online and offline data repair and cleaning on the edge and in the cloud.
(f) The programming languages and solutions for managing data quality (e.g., TensorFlow, Keras, MATLAB's fuzzy logic toolbox) highly depend on the solution domain (e.g., ML, data mining). Big data software platforms are suitable for addressing data quality concerns for high-velocity data (e.g., the timely monitoring of data changes, data redundancy, data inconsistency, and data insecurity).
(g) We could not reveal any direct relation between the database technologies and the data quality techniques. On the other hand, big data systems gathering data from various sources combine multiple data storage solutions that require a layered data storage architecture where each layer may require its own data quality technique.

ACKNOWLEDGMENTS

We would like to thank Enrique Garcia-Ceja (our former colleague at SINTEF), Nicolas Jourdan (PTW TU Darmstadt, Germany), and Beatriz Bretones Cassoli (PTW TU Darmstadt, Germany) for their help during the initial phase of this work. This work has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No. 958357 (InterQ), and Grant Agreement No. 958363 (DAT4.Zero).

PRIMARY STUDIES

[1] Mohamed Ahmed, Chantal Taconet, Mohamed Ould, Sophie Chabridon, and Amel Bouzeghoub. 2021. IoT Data Qualification for a Logistic Chain Traceability Smart Contract. *Sensors* 21, 6 (2021).
[2] Daniele Apiletti, Claudia Barberis, Tania Cerquitelli, Alberto Macii, Enrico Macii, Massimo Poncino, and Francesco Ventura. 2018. iSTEP, an Integrated Self-Tuning Engine for Predictive Maintenance in Industry 4.0. In *ISPA/IUCC/BDCloud/SocialCom/SustainCom'18*. 924–931.
[3] Shelernaz Azimi and Claus Pahl. 2020. A Layered Quality Framework for Machine Learning-driven Data and Information Models.. In *ICEIS (1)*. 579–587.
[4] Oluwaseun Bamgboye, Xiaodong Liu, and Peter Cruickshank. 2019. Semantic Stream Management Framework for Data Consistency in Smart Spaces. In *COMPSAC'19*. 85–90.
[5] Sebastian Buschjäger and Katharina Morik. 2018. Decision Tree and Random Forest Implementations for Fast Filtering of Sensor Data. *IEEE Transactions on Circuits and Systems I: Regular Papers* 65, 1 (2018), 209–222.
[6] John Byabazaire, Gregory O'Hare, and Declan Delaney. 2020. Using Trust as a Measure to Derive Data Quality in Data Shared IoT Deployments. In *ICCCN'20*. 1–9.

[7] Roberto Casado-Vara, Fernando de la Prieta, Javier Prieto, and Juan M. Corchado. 2018. Blockchain Framework for IoT Data Quality via Edge Computing. In *BlockSys'18*. 19–24.
[8] T. Cerquitelli, N. Nikolakis, P. Bethaz, S. Panicucci, F. Ventura, E. Macii, S. Andolina, A. Marguglio, K. Alexopoulos, P. Petrali, A. Pagani, P. van Wilgen, and M. Ippolito. 2020. Enabling predictive analytics for smart manufacturing through an IIoT platform. In *AMEST'20*. 179–184.
[9] David Camilo Corrales, Juan Carlos Corrales, and Agapito Ledezma. 2018. How to Address the Data Quality Issues in Regression Models: A Guided Process for Data Cleaning. *Symmetry* 10, 4 (2018).
[10] Yesheng Cui, Sami Kara, and Ka C. Chan. 2020. Monitoring and Control of Unstructured Manufacturing Big Data. In *IEEM'20*. 928–932.
[11] Volkan Dedeoglu, Raja Jurdak, Guntur D. Putra, Ali Dorri, and Salil S. Kanhere. 2019. A Trust Architecture for Blockchain in IoT. In *MobiQuitous'19*. 190–199.
[12] Dominik Flick, Sebastian Gellrich, Marc-André Filz, Li Ji, Sebastian Thiede, and Christoph Herrmann. 2019. Conceptual Framework for manufacturing data preprocessing of diverse input sources. In *INDIN'19*. 1041–1046.
[13] Ziqi Guo, Tingwen Bao, Wenlong Wu, Chao Jin, and Jay Lee. 2019. IAI DevOps: A Systematic Framework for Prognostic Model Lifecycle Management. In *PHM-Qingdao'19*. 1–6.
[14] Liu Hui, Ye Xiaobo, Meng Zhijun, Zhou Lijuan, and Sun Zhong. 2019. An Agricultural Machinery Operation Monitoring System Based on IoT. In *DSIT'19*. 225–229.
[15] Dan Huru, Cătălin Leordeanu, Elena Apostol, Mariana Mocanu, and Valentin Cristea. 2018. BigClue Analytics: A Middleware Component for Modeling Sensor Data in IoT Systems. In *HPCC/SmartCity/DSS'18*. 891–896.
[16] Seunghwan Jeong, Gwangpyo Yoo, Minjong Yoo, Ikjun Yeom, and Honguk Woo. 2019. Resource-Efficient Sensor Data Management for Autonomous Systems Using Deep Reinforcement Learning. *Sensors* 19, 20 (2019).
[17] Thomas Küfner, Stefan Schönig, Richard Jasinski, and Andreas Ermer. 2021. Vertical data continuity with lean edge analytics for industry 4.0 production. *Computers in Industry* 125 (2021), 103389.
[18] Mohammad Ayoub Khan and Fahad Algarni. 2020. A Healthcare Monitoring System for the Diagnosis of Heart Disease in the IoMT Cloud Environment Using MSSO-ANFIS. *IEEE Access* 8 (2020), 122259–122269.
[19] Tianxiang Kong, Tianliang Hu, Tingting Zhou, and Yingxin Ye. 2021. Data Construction Method for the Applications of Workshop Digital Twin System. *Journal of Manufacturing Systems* 58 (2021), 323–328.
[20] Marina Krotofil, Jason Larsen, and Dieter Gollmann. 2015. The Process Matters: Ensuring Data Veracity in Cyber-Physical Systems. In *ASIA CCS'15*. 133–144.
[21] Daniel Kuemper, Thorben Iggena, Ralf Toenjes, and Elke Pulvermueller. 2018. Valid.IoT: A Framework for Sensor Data Quality Analysis and Interpolation. In *MMSys'18*. 294–303.
[22] Fei Li, Stefan Nastic, and Schahram Dustdar. 2012. Data Quality Observation in Pervasive Environments. In *ICCSE'12*. 602–609.
[23] Wei-Tsung Lin, Fatih Bakir, Chandra Krintz, Rich Wolski, and Markus Mock. 2019. Data Repair for Distributed, Event-Based IoT Applications. In *DEBS'19*. 139–150.
[24] Chao Liu, Léopold Le Roux, Carolin Körner, Olivier Tabaste, Franck Lacan, and Samuel Bigot. 2020. Digital Twin-enabled Collaborative Data Management for Metal Additive Manufacturing Systems. *Journal of Manufacturing Systems* (2020).
[25] Carina Mieth, Anne Meyer, and Michael Henke. 2019. Framework for the usage of data from real-time indoor localization systems to derive inputs for manufacturing simulation. *Procedia CIRP* 81 (2019), 868–873.
[26] Fatma Mohammed, A. S. M. Kayes, Eric Pardede, and Wenny Rahayu. 2020. A Framework for Measuring IoT Data Quality Based on Freshness Metrics. In *TrustCom'20*. 1242–1249.
[27] Nwamaka U. Okafor, Yahia Alghorani, and Declan T. Delaney. 2020. Improving Data Quality of Low-cost IoT Sensors in Environmental Monitoring Networks Using Data Fusion and Machine Learning Approach. *ICT Express* 6, 3 (2020), 220–228.
[28] Michael S Packianather, Nury Leon Munizaga, Soha Zouwail, and Mark Saunders. 2019. Development of soft computing tools and IoT for improving the performance assessment of analysers in a clinical laboratory. In *SoSE'19*. 158–163.
[29] Stefano Proto, Francesco Ventura, Daniele Apiletti, Tania Cerquitelli, Elena Baralis, Enrico Macii, and Alberto Macii. 2019. PREMISES, a Scalable Data-Driven Service to Predict Alarms in Slowly-Degrading Multi-Cycle Industrial Processes. In *BigDataCongress'19*. 139–143.
[30] Qinglin Qi and Fei Tao. 2018. Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. *IEEE Access* 6 (2018), 3585–3593.
[31] Luke Russell, Felix Kwamena, and Rafik Goubran. 2019. Towards Reliable IoT: Fog-Based AI Sensor Validation. In *IEEE Cloud Summit*. 37–44.
[32] Maribel Yasmina Santos, Jorge Oliveira e Sá, Carina Andrade, Francisca Vale Lima, Eduarda Costa, Carlos Costa, Bruno Martinho, and João Galvão. 2017. A Big Data system supporting Bosch Braga Industry 4.0 strategy. *International Journal of Information Management* 37, 6 (2017), 750–760.
[33] Sunny Sanyal and Puning Zhang. 2018. Improving Quality of Data: IoT Data Aggregation Using Device to Device Communications. *IEEE Access* 6 (2018), 67830–67840.

[34] E Saranya and P. Bagavathi Sivakumar. 2020. Data-Driven Prognostics for Run-To-Failure Data Employing Machine Learning Models. In *ICICT'20*. 528–533.
[35] Mahmoud Reza Saybani, Teh Ying Wah, Amineh Amini, and Saeed Reza Aghabozorgi Sahaf Yazdi. 2011. Anomaly detection and prediction of sensors faults in a refinery using data mining techniques and fuzzy logic. *Scientific Research and Essays* 6, 27 (2011), 5685–5695.
[36] Elena Seghezzi, Mirko Locatelli, Laura Pellegrini, Giulia Pattini, Giuseppe Martino Di Giuda, Lavinia Chiara Tagliabue, and Guido Boella. 2021. Towards an Occupancy-Oriented Digital Twin for Facility Management: Test Campaign and Sensors Assessment. *Applied Sciences* 11, 7 (2021).
[37] Devarshi Shah, Jin Wang, and Q. Peter He. 2020. Feature engineering in big data analytics for IoT-enabled smart manufacturing – Comparison between deep learning and statistical learning. *Computers & Chemical Engineering* 141 (2020), 106970.
[38] Sabrina Sicari, Cinzia Cappiello, Francesco De Pellegrini, Daniele Miorandi, and Alberto Coen-Porisini. 2016. A security-and quality-aware system architecture for Internet of Things. *Information Systems Frontiers* 18, 4 (2016), 665–677.
[39] Sabrina Sicari, Alessandra Rizzardi, Daniele Miorandi, Cinzia Cappiello, and Alberto Coen-Porisini. 2016. A secure and quality-aware prototypical architecture for the Internet of Things. *Information Systems* 58 (2016), 43–55.
[40] Muhammad Syafrudin, Ganjar Alfian, Norma Latif Fitriyani, and Jongtae Rhee. 2018. Performance Analysis of IoT-Based Sensor, Big Data Processing, and Machine Learning Model for Real-Time Monitoring System in Automotive Manufacturing. *Sensors* 18, 9 (2018).
[41] Chen-Khong Tham and Rajalaxmi Rajagopalan. 2020. Active Learning for IoT Data Prioritization in Edge Nodes Over Wireless Networks. In *IECON'20*. 4453–4458.
[42] K. Villalobos, V.J. Ramírez-Durán, B. Diez, J.M. Blanco, A. Goñi, and A. Illarramendi. 2020. A three level hierarchical architecture for an efficient storage of industry 4.0 data. *Computers in Industry* 121 (2020), 103257.
[43] Kevin Villalobos, Jon Vadillo, Borja Diez, Borja Calvo, and Arantza Illarramendi. 2018. I4TSPS: a visual-interactive web system for industrial time-series pre-processing. In *Big Data'18*. 2012–2018.
[44] Chang Wang, Yongxin Zhu, Weiwei Shi, Victor Chang, P. Vijayakumar, Bin Liu, Yishu Mao, Jiabao Wang, and Yiping Fan. 2018. A Dependable Time Series Analytic Framework for Cyber-Physical Systems of IoT-Based Smart Grid. *ACM Transactions on Cyber-Physical Systems* 3, 1 (2018), 18 pages.
[45] Wei Wei, Jun Yuan, and Ang Liu. 2020. Manufacturing data-driven process adaptive design method. *Procedia CIRP* 91 (2020), 728–734.
[46] Sholom M. Weiss, Amit Dhurandhar, and Robert J. Baseman. 2013. Improving Quality Control by Early Prediction of Manufacturing Outcomes. In *KDD'13*. 1258–1266.
[47] Leon Wu and Gail Kaiser. 2012. An Autonomic Reliability Improvement System for Cyber-Physical Systems. In *HASE'12*. 56–61.
[48] Wenjin Yu, Tharam Dillon, Fahed Mostafa, Wenny Rahayu, and Yuehua Liu. 2019. Implementation of Industrial Cyber Physical System: Challenges and Solutions. In *ICPS'19*. 173–178.
[49] Wenjin Yu, Tharam Dillon, Fahed Mostafa, Wenny Rahayu, and Yuehua Liu. 2020. A Global Manufacturing Big Data Ecosystem for Fault Detection in Predictive Maintenance. *IEEE Transactions on Industrial Informatics* 16, 1 (2020), 183–192.
[50] Alejandro Gabriel Villanueva Zacarias, Peter Reimann, and Bernhard Mitschang. 2018. A framework to guide the selection and configuration of machine-learning-based data analytics solutions in manufacturing. *Procedia CIRP* 72 (2018), 153–158.
[51] Werner Zellinger, Volkmar Wieser, Mohit Kumar, David Brunner, Natalia Shepeleva, Rafa Gálvez, Josef Langer, Lukas Fischer, and Bernhard Moser. 2021. Beyond federated learning: On confidentiality-critical machine learning applications in industry. In *ISM'20*. 734–743.

REFERENCES

[52] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *OSDI'16*. 265–283.
[53] Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* 2, 4 (2010), 433–459.
[54] Rasheed Ahmad and Izzat Alsmadi. 2021. Machine learning approaches to IoT security: A systematic literature review. *Internet of Things* 14 (2021), 100365.
[55] Shahriar Akter, Grace McCarthy, Shahriar Sajib, Katina Michael, Yogesh K Dwivedi, John D'Ambra, and KN Shen. 2021. Algorithmic bias in data-driven innovation in the age of AI. *International Journal of Information Management* 60 (2021), 102387.
[56] Mohammed Ali Al-Garadi, Amr Mohamed, Abdulla Khalid Al-Ali, Xiaojiang Du, Ihsan Ali, and Mohsen Guizani. 2020. A survey of machine and deep learning methods for internet of things (IoT) security. *IEEE Communications Surveys & Tutorials* 22, 3 (2020), 1646–1685.
[57] Iqbal Alam, Kashif Sharif, Fan Li, Zohaib Latif, Md Monjurul Karim, Sujit Biswas, Boubakr Nour, and Yu Wang. 2020. A survey of network virtualization techniques for Internet of Things using SDN and NFV. *ACM Computing Surveys (CSUR)* 53, 2 (2020), 1–40.

[58] Ahmed Abdulhasan Alwan, Mihaela Anca Ciupala, Allan J Brimicombe, Seyed Ali Ghorashi, Andres Baravalle, and Paolo Falcarin. 2022. Data quality challenges in large-scale cyber-physical systems: A systematic review. *Information Systems* 105 (2022).
[59] Mahmoud Ammar, Giovanni Russello, and Bruno Crispo. 2018. Internet of Things: A survey on the security of IoT frameworks. *Journal of Information Security and Applications* 38 (2018), 8–27.
[60] Hamidreza Arasteh, Vahid Hosseinnezhad, Vincenzo Loia, Aurelio Tommasetti, Orlando Troisi, Miadreza Shafie-khah, and Pierluigi Siano. 2016. Iot-based smart cities: A survey. In *2016 IEEE 16th international conference on environment and electrical engineering (EEEIC)*. 1–6.
[61] Parvaneh Asghari, Amir Masoud Rahmani, and Hamid Haj Seyyed Javadi. 2019. Internet of Things applications: A systematic review. *Computer Networks* 148 (2019), 241–261.
[62] Moslem Azamfar, Xiang Li, and Jay Lee. 2020. Deep learning-based domain adaptation method for fault diagnosis in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing* 33, 3 (2020), 445–453.
[63] Maggi Bansal, Inderveer Chana, and Siobhán Clarke. 2020. A survey on iot big data: current status, 13 v's challenges, and future directions. *ACM Computing Surveys (CSUR)* 53, 6 (2020), 1–59.
[64] Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. 2009. C-SPARQL: SPARQL for continuous querying. In *WWW'09*. 1061–1062.
[65] Dina Bitton and David J DeWitt. 1983. Duplicate record elimination in large data files. *ACM Transactions on database systems* 8, 2 (1983), 255–265.
[66] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *MOD'00*. 93–104.
[67] Michael Cafarella, Ihab F Ilyas, Marcel Kornacker, Tim Kraska, and Christopher Ré. 2016. Dark Data: Are we solving the right problems?. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 1444–1445.
[68] Apache Camel. [n. d.]. https://camel.apache.org/.
[69] Jeremy J Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. 2004. Jena: implementing the semantic web recommendations. In *WWW'04*. 74–83.
[70] Z Berkay Celik, Earlence Fernandes, Eric Pauley, Gang Tan, and Patrick McDaniel. 2019. Program analysis of commodity IoT applications for security and privacy: Challenges and opportunities. *ACM Computing Surveys (CSUR)* 52, 4 (2019), 1–30.
[71] Hong Chen. 2017. Applications of cyber-physical system: a literature review. *Journal of Industrial Integration and Management* 2, 03 (2017), 1750012.
[72] Zhiyan Chen, Jinxin Liu, Yu Shen, Murat Simsek, Burak Kantarci, Hussein T Mouftah, and Petar Djukic. 2022. Machine learning-enabled iot security: Open issues and challenges under advanced persistent threats. *Comput. Surveys* 55, 5 (2022), 1–37.
[73] Lalit Chettri and Rabindranath Bera. 2019. A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems. *IEEE Internet of Things Journal* 7, 1 (2019), 16–32.
[74] Angelo Corallo, Anna Maria Crespino, Vito Del Vecchio, Mariangela Lazoi, and Manuela Marra. 2021. Understanding and Defining Dark Data for the Manufacturing Industry. *IEEE Transactions on Engineering Management* (2021).
[75] Create-IoT. 2018. Deliverable D6.02 – Recommendations for commonalities and interoperability profiles of IoT platforms. https://european-iot-pilots.eu/wp-content/uploads/2018/11/D06_02_WP06_H2020_CREATE-IoT_Final.pdf
[76] Li Da Xu, Wu He, and Shancang Li. 2014. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics* 10, 4 (2014), 2233–2243.
[77] Nilanjan Dey, Amira S Ashour, Fuqian Shi, Simon James Fong, and João Manuel RS Tavares. 2018. Medical cyber-physical systems: A survey. *Journal of medical systems* 42 (2018), 1–13.
[78] Jasenka Dizdarević, Francisco Carpio, Admela Jukan, and Xavi Masip-Bruin. 2019. A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration. *ACM Computing Surveys (CSUR)* 51, 6 (2019), 1–29.
[79] Alan F Dutka and Howard H Hansen. 1991. *Fundamentals of data normalization.* Addison-Wesley Longman Publishing Co., Inc.
[80] Olakunle Elijah, Tharek Abdul Rahman, Igbafe Orikumhi, Chee Yen Leow, and MHD Nour Hindia. 2018. An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet of things Journal* 5, 5 (2018), 3758–3773.
[81] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96*, Vol. 96. 226–231.
[82] Alejandro Germán Frank, Lucas Santos Dalenogare, and Néstor Fabián Ayala. 2019. Industry 4.0 technologies: Implementation patterns in manufacturing companies. *International Journal of Production Economics* 210 (2019), 15–26.
[83] Gregory Gimpel and Allan Alter. 2021. Benefit From the Internet of Things Right Now by Accessing Dark Data. *IT Professional* 23, 2 (2021), 45–49.
[84] Jairo Giraldo, Esha Sarkar, Alvaro A Cardenas, Michail Maniatakos, and Murat Kantarcioglu. 2017. Security and privacy in cyber-physical systems: A survey of surveys. *IEEE Design & Test* 34, 4 (2017), 7–17.
[85] Jairo Giraldo, David Urbina, Alvaro Cardenas, Junia Valente, Mustafa Faisal, Justin Ruths, Nils Ole Tippenhauer, Henrik Sandberg, and Richard Candell. 2018. A survey of physics-based attack detection in cyber-physical systems. *ACM Computing Surveys (CSUR)* 51, 4

(2018), 1–36.
[86] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Machine learning basics. *Deep learning* 1, 7 (2016), 98–164.
[87] Emily Grantner. 2007. ISO 8000: a standard for data quality. *Logistics Spectrum* 41, 4 (2007).
[88] Venkat N Gudivada, Srini Ramaswamy, and Seshadri Srinivasan. 2018. Data management issues in cyber-physical systems. In *Transportation Cyber-Physical Systems*. 173–200.
[89] Antonio Gulli and Sujit Pal. 2017. *Deep learning with Keras*. Packt Publishing Ltd.
[90] Volkan Gunes, Steffen Peter, Tony Givargis, and Frank Vahid. 2014. A survey on concepts, applications, and challenges in cyber-physical systems. *KSII Transactions on Internet and Information Systems (TIIS)* 8, 12 (2014), 4242–4268.
[91] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11, 1 (2009), 10–18.
[92] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. 2019. Differential privacy techniques for cyber physical systems: a survey. *IEEE Communications Surveys & Tutorials* 22, 1 (2019), 746–789.
[93] Wan Haslina Hassan et al. 2019. Current research on Internet of Things (IoT) security: A survey. *Computer networks* 148 (2019), 283–294.
[94] Vikas Hassija, Vinay Chamola, Vikas Saxena, Divyansh Jain, Pranav Goyal, and Biplab Sikdar. 2019. A survey on IoT security: application areas, security threats, and solution architectures. *IEEE Access* 7 (2019), 82721–82743.
[95] Abhishek Hazra, Mainak Adhikari, Tarachand Amgoth, and Satish Narayana Srirama. 2021. A comprehensive survey on interoperability for IIoT: taxonomy, standards, and future directions. *ACM Computing Surveys (CSUR)* 55, 1 (2021), 1–35.
[96] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li, and Bo Luo. 2017. Cyber-physical systems security—A survey. *IEEE Internet of Things Journal* 4, 6 (2017), 1802–1831.
[97] DNV International. 2017. *Data quality assessment framework: DNV Recommended Practice-RP-0497*. DNV.
[98] Juxtology. 2018. IoT: Architecture. https://www.m2mology.com/iot-transformation/iot-world-forum/
[99] Apache Kafka. [n. d.]. https://kafka.apache.org/.
[100] Aimad Karkouch, Hajar Mousannif, Hassan Al Moatassime, and Thomas Noel. 2016. Data quality in internet of things: A state-of-the-art survey. *Journal of Network and Computer Applications* 73 (2016), 57–81.
[101] Mostafa Haghi Kashani, Mona Madanipour, Mohammad Nikravan, Parvaneh Asghari, and Ebrahim Mahdipour. 2021. A systematic review of IoT in healthcare: Applications, techniques, and trends. *Journal of Network and Computer Applications* 192 (2021), 103164.
[102] Hakan Kayan, Matthew Nunes, Omer Rana, Pete Burnap, and Charith Perera. 2022. Cybersecurity of industrial cyber-physical systems: a review. *ACM Computing Surveys (CSUR)* 54, 11s (2022), 1–35.
[103] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. 2014. A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference*. 372–378.
[104] Barbara Ann Kitchenham and Stuart Charters. 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Technical Report EBSE 2007-001. https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf
[105] Laphou Lao, Zecheng Li, Songlin Hou, Bin Xiao, Songtao Guo, and Yuanyuan Yang. 2020. A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling. *ACM Computing Surveys (CSUR)* 53, 1 (2020), 1–32.
[106] Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788–791.
[107] Shancang Li, Li Da Xu, and Shanshan Zhao. 2018. 5G Internet of Things: A survey. *Journal of Industrial Information Integration* 10 (2018), 1–9.
[108] Shancang Li, Li Da Xu, and Shanshan Zhao. 2015. The internet of things: a survey. *Information systems frontiers* 17 (2015), 243–259.
[109] Yongxin Liao, Fernando Deschamps, Eduardo de Freitas Rocha Loures, and Luiz Felipe Pierin Ramos. 2017. Past, present and future of Industry 4.0-a systematic literature review and research agenda proposal. *International journal of production research* 55, 12 (2017), 3609–3629.
[110] Caihua Liu, Patrick Nitschke, Susan P Williams, and Didar Zowghi. 2020. Data quality and the Internet of Things. *Computing* 102, 2 (2020), 573–599.
[111] Yang Lu. 2017. Industry 4.0: A survey on technologies, applications and open research issues. *Journal of industrial information integration* 6 (2017), 1–10.
[112] Yuriy Zacchia Lun, Alessandro D'Innocenzo, Francesco Smarra, Ivano Malavolta, and Maria Domenica Di Benedetto. 2019. State of the art of cyber-physical systems security: An automatic control perspective. *Journal of Systems and Software* 149 (2019), 174–216.
[113] Yuan Luo, Ya Xiao, Long Cheng, Guojun Peng, and Danfeng Yao. 2021. Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–36.
[114] Benjamin Maschler, Hannes Vietz, Nasser Jazdi, and Michael Weyrich. 2020. Continual learning of fault prediction for turbofan engines using deep learning with elastic weight consolidation. In *ETFA'20*. 959–966.
[115] Sara N Matheu, Jose L Hernandez-Ramos, Antonio F Skarmeta, and Gianmarco Baldini. 2020. A survey of cybersecurity certification for the Internet of Things. *ACM Computing Surveys (CSUR)* 53, 6 (2020), 1–36.

[116] MathWorks Matlab. [n. d.]. https://mathworks.com/products/matlab.html.
[117] Francesca Meneghello, Matteo Calore, Daniel Zucchetto, Michele Polese, and Andrea Zanella. 2019. IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet of Things Journal* 6, 5 (2019), 8182–8201.
[118] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, et al. 2016. Mllib: Machine learning in apache spark. *Journal of Machine Learning Research* 17, 1 (2016), 1235–1241.
[119] Robert Mitchell and Ing-Ray Chen. 2014. A survey of intrusion detection techniques for cyber-physical systems. *ACM Computing Surveys (CSUR)* 46, 4 (2014), 1–29.
[120] Supavit Muangjaroen and Thaweesak Yingthawornsuk. 2012. A study of noise reduction in speech signal using fir filtering. In *International Conference on Advances in Electrical and Electronics Engineering*.
[121] Bruce Jay Nelson. 1981. *Remote procedure call.* Carnegie Mellon University.
[122] Nataliia Neshenko, Elias Bou-Harb, Jorge Crichigno, Georges Kaddoum, and Nasir Ghani. 2019. Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations. *IEEE Communications Surveys & Tutorials* 21, 3 (2019), 2702–2733.
[123] Anne H Ngu, Mario Gutierrez, Vangelis Metsis, Surya Nepal, and Quan Z Sheng. 2016. IoT middleware: A survey on issues and enabling technologies. *IEEE Internet of Things Journal* 4, 1 (2016), 1–20.
[124] Phu H Nguyen, Shaukat Ali, and Tao Yue. 2017. Model-based security engineering for cyber-physical systems: A systematic mapping study. *Information and Software Technology* 83 (2017), 116–135.
[125] Apache NiFi. [n. d.]. https://nifi.apache.org/.
[126] K Nose-Filho, ADP Lotufo, and CR Minussi. 2011. Preprocessing data for short-term load forecasting with a general regression neural network and a moving average filter. In *IEEE Trondheim PowerTech*. 1–7.
[127] Sophocles J Orfanidis. 2016. *Introduction to signal processing.* Pearson Education, Inc.
[128] Ercan Oztemel and Samet Gursev. 2020. Literature review of Industry 4.0 and related technologies. *Journal of intelligent manufacturing* 31 (2020), 127–182.
[129] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[130] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology* 64 (2015), 1–18.
[131] Apache Phoenix. [n. d.]. https://phoenix.apache.org/.
[132] The Node.js platform. [n. d.]. https://nodejs.org/en/.
[133] Ioannis Prapas, Behrouz Derakhshan, Alireza Rezaei Mahdiraji, and Volker Markl. 2021. Continuous Training and Deployment of Deep Learning Models. *Datenbank-Spektrum* 21, 3 (2021), 203–212.
[134] NMEA 0183 Protocol. [n. d.]. https://www.nmea.org/content/STANDARDS/NMEA_0183_Standard.
[135] REST. [n. d.]. https://restfulapi.net/.
[136] Manuel Sanchez, Ernesto Exposito, and Jose Aguilar. 2020. Industry 4.0: survey from a system integration perspective. *International Journal of Computer Integrated Manufacturing* 33, 10-11 (2020), 1017–1041.
[137] Sajjad Hussain Shah and Ilyas Yaqoob. 2016. A survey: Internet of Things (IOT) technologies, applications and challenges. *2016 IEEE Smart Energy Grid Engineering (SEGE)* (2016), 381–385.
[138] Dudley Shapere. 1964. The structure of scientific revolutions. *The Philosophical Review* 73, 3 (1964), 383–394.
[139] Weiwei Shi, Yongxin Zhu, Jinkui Zhang, Xiang Tao, Gehao Sheng, Yong Lian, Guoxing Wang, and Yufeng Chen. 2015. Improving power grid monitoring data quality: An efficient machine learning framework for missing data prediction. In *HPCC/CSS/ICESS'15*. 417–422.
[140] Jeffrey S Simonoff. 2012. *Smoothing methods in statistics.* Springer Science & Business Media.
[141] Eugene Siow, Thanassis Tiropanis, and Wendy Hall. 2018. Analytics for the internet of things: A survey. *ACM computing surveys (CSUR)* 51, 4 (2018), 1–36.
[142] Apache Spark. [n. d.]. https://spark.apache.org/.
[143] Muhammad Talha, Anas Abou El Kalam, and Nabil Elmarzouqi. 2019. Big data: Trade-off between data quality and data security. *Procedia Computer Science* 151 (2019), 916–922.
[144] Hui Yie Teh, Andreas W. Kempa-Liehr, and Kevin I-Kai Wang. 2020. Sensor data quality: a systematic review. *Journal of Big Data* 7, 1 (2020), 11.
[145] Nguyen Khoi Tran, Quan Z Sheng, Muhammad Ali Babar, and Lina Yao. 2017. Searching the Web OF Things: state of the art, challenges, and solutions. *ACM Computing Surveys (CSUR)* 50, 4 (2017), 55.
[146] Nazar Waheed, Xiangjian He, Muhammad Ikram, Muhammad Usman, Saad Sajid Hashmi, and Muhammad Usman. 2020. Security and privacy in IoT using machine learning and blockchain: Threats and countermeasures. *ACM Computing Surveys (CSUR)* 53, 6 (2020), 1–37.

[147] Richard Y. Wang and Diane M. Strong. 1996. Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems* 12, 4 (1996), 5–33.
[148] Xi Wang and Chen Wang. 2019. Time series data cleaning: A survey. *IEEE Access* 8 (2019), 1866–1881.
[149] Y Richard Wang, Lisa M Guarascio, and Richard Wang. 1991. Dimensions of data quality: Toward quality data by design. (1991).
[150] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *EASE'14*. 38.
[151] Rich Wolski, Chandra Krintz, Fatih Bakir, Gareth George, and Wei-Tsung Lin. 2019. CSPOT: Portable, Multi-scale Functions-as-a-service for IoT. In *SEC'19*. 236–249.
[152] Hui Xiong, Gaurav Pandey, Michael Steinbach, and Vipin Kumar. 2006. Enhancing data analysis with noise removal. *IEEE Transactions on Knowledge and Data Engineering* 18, 3 (2006), 304–319.
[153] Hansong Xu, Wei Yu, David Griffith, and Nada Golmie. 2018. A survey on industrial Internet of Things: A cyber-physical systems perspective. *IEEE access* 6 (2018), 78238–78259.
[154] Li Da Xu and Lian Duan. 2019. Big data for cyber physical systems in industry 4.0: a survey. *Enterprise Information Systems* 13, 2 (2019), 148–169.
[155] Li Da Xu, Eric L Xu, and Ling Li. 2018. Industry 4.0: state of the art and future trends. *International journal of production research* 56, 8 (2018), 2941–2962.
[156] Lina Zhang, Dongwon Jeong, and Sukhoon Lee. 2021. Data Quality Management in the Internet of Things. *Sensors* 21, 17 (2021), 5834.
[157] Ting Zheng, Marco Ardolino, Andrea Bacchetti, and Marco Perona. 2021. The applications of Industry 4.0 technologies in manufacturing context: a systematic literature review. *International Journal of Production Research* 59, 6 (2021), 1922–1954.
[158] Qingyi Zhu, Seng W Loke, Rolando Trujillo-Rasua, Frank Jiang, and Yong Xiang. 2019. Applications of distributed ledger technologies to the internet of things: A survey. *ACM computing surveys (CSUR)* 52, 6 (2019), 1–34.