Journal of Big Data

SURVEY PAPER Open Access

Sensor data quality: a systematic review

Hui Yie Teh[1], Andreas W. Kempa-Liehr[2,3*] and Kevin I-Kai Wang[1]

*Correspondence: kempa-liehr@fmf.uni-freiburg.de
[2] Freiburg Materials Research Center, University of Freiburg, Freiburg, Germany
Full list of author information is available at the end of the article

Abstract

Sensor data quality plays a vital role in Internet of Things (IoT) applications as they are rendered useless if the data quality is bad. This systematic review aims to provide an introduction and guide for researchers who are interested in quality-related issues of physical sensor data. The process and results of the systematic review are presented which aims to answer the following research questions: what are the different types of physical sensor data errors, how to quantify or detect those errors, how to correct them and what domains are the solutions in. Out of 6970 literatures obtained from three databases (ACM Digital Library, IEEE Xplore and ScienceDirect) using the search string refined via topic modelling, 57 publications were selected and examined. Results show that the different types of sensor data errors addressed by those papers are mostly missing data and faults e.g. outliers, bias and drift. The most common solutions for error detection are based on principal component analysis (PCA) and artificial neural network (ANN) which accounts for about 40% of all error detection papers found in the study. Similarly, for fault correction, PCA and ANN are among the most common, along with Bayesian Networks. Missing values on the other hand, are mostly imputed using Association Rule Mining. Other techniques include hybrid solutions that combine several data science methods to detect and correct the errors. Through this systematic review, it is found that the methods proposed to solve physical sensor data errors cannot be directly compared due to the non-uniform evaluation process and the high use of non-publicly available datasets. Bayesian data analysis done on the 57 selected publications also suggests that publications using publicly available datasets for method evaluation have higher citation rates.

Keywords: Systematic review, Sensor data quality, Sensor data error detection, Sensor data error correction, Datasets

Introduction

With the emergence of the Internet of Things (IoT) and wireless sensor networks (WSNs), sensor devices are deployed across the globe in a variety of fields such as healthcare, industry, agriculture, home, and transport [1]. Recently, Cisco [2] estimated that there would be approximately 850 zettabytes (1 zettabyte is 10^{21} bytes) of data generated from devices. An IoT application may have hundreds or thousands of sensors which produces vast amounts of data, but the data is rendered useless if it is riddled with errors as poor sensor data quality caused by the errors may lead to wrong decision-making results. In this paper, the term *sensor* refers to a physical sensor [3, Chap 3], which measures the changes in physical quantity e.g. temperature, humidity, and light

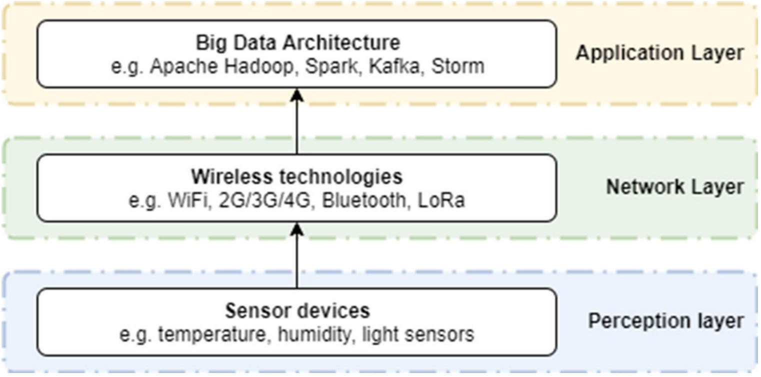
Big Data Architecture
e.g. Apache Hadoop, Spark, Kafka, Storm
Application Layer
Wireless technologies
e.g. WiFi, 2G/3G/4G, Bluetooth, LoRa
Network Layer
Sensor devices
e.g. temperature, humidity, light sensors
Perception layer

Fig. 1 IoT architecture. A high-level view of a typical IoT architecture which shows the data flow

intensity of the sample or surroundings. Furthermore, the term *error* relates to the soft faults that occur in sensor data found commonly in the systematic review such as outliers, bias, drifts, missing values, and uncertainty, which should be detected or quantified and removed or corrected in order to improve sensor data quality.

This is slightly different from the data quality (DQ) dimensions introduced by Wang and Strong [4], which categorize the quality of data in databases or high-level application architecture (Application Layer in Fig. 1) that are important to data consumers. They are mostly used to describe data in enterprise-level systems and are used for modelling how data errors propagate to the consumer's end. Therefore, apart from incomplete (missing data) and inaccurate data (uncertainty), which are sensor data quality-related issues, other DQ dimensions such as inconsistent data and timeliness are not considered in this review paper as they are more specific to the topics of database design or communication data quality. A survey related to DQ dimensions is presented in the works of Karkouch et al. [5].

Figure 1 shows an overview of the data flow of a typical IoT application. A physical sensor such as a temperature or humidity sensor measures and collects readings (changes in the observed property) in the *Perception layer*. The readings are then transmitted through the *Network layer*, which determines the routes to send the sensor data and is implemented using wireless technologies such as WiFi, 2G/3G/4G, Bluetooth, and LoRa. Next, the *Application layer* receives data from the network layer and it is where the data processing, predictive analytics [6], and storage takes place. The application layer is designed and implemented using big data architectures such as Apache Hadoop, Spark, or Kafka. The added complexity of the architecture causes new errors to be potentially introduced in each layer. For example, in the Network Layer, poor data quality arises from congested and unstable wireless communication links in sensor networks which causes data loss and corruption [7]. In the Perception Layer, damage or exhaustion of battery in sensor devices also causes data quality to degrade, as towards the end of its battery life, sensors tend to produce unstable readings [8]. The hostile environment in which in-situ sensors are deployed also plays a big part in the quality of the transmitted data. For example, sensors for temperature, light, or humidity measurements are often placed outdoors and are subjected to extreme local weather conditions such as strong winds and snow, which might affect the operation of the sensor.

Although the factors that cause errors and affect sensor data quality are known, simple strategies to overcome data quality problems, such as using industry grade sensors, which are more accurate, stable and robust, are not feasible for applications that require the deployment of large and dense sensors networks, which is the case for many IoT applications. For example, in horticulture, sensors need to be deployed such that they have high coverage and accuracy through large and dense sensor networks. Having to deploy many highly accurate but expensive sensors will incur higher deployment costs. Therefore, most IoT applications use low-cost sensors, though at the expense of data quality. The use of both industry grade or low-cost sensors also results in high time and maintenance cost as experts would have to go out to the field themselves to test and calibrate the entire network of in-situ sensors to ensure data quality. Other than that, re-transmitting the data when experiencing data quality errors (e.g. missing data) also does not work well in an IoT application. This is because the nodes in the network are powered on limited battery and memory which makes it expensive in terms of power and computational resources to resend the missing data across the network, especially if there is a big load of data to re-transmit. Retransmission also delays decision-making which in turn may lead to inaccurate results [9].

Other than that, though studies in previous years tend to focus on high-level solutions in the Application Layer for solving data quality issues [4, 10], it is not possible nowadays due to the separation of the layers and complexity of the architecture. The advance of Big Data where the sheer volume of data hinders the transport to the central system [1] also encourages edge computing, or a decentralized solution, where the processing of data quality is done in the Perception Layer i.e. in the sensor devices themselves and only data with good quality is passed to the central server. Since sensor data errors may be present and propagated in all layers, this review paper focuses on algorithms that solves the fundamental issue of sensor data quality by detecting and correcting those errors regardless of the IoT (or big data) architecture and layers. As such, the high-level design and decision of the IoT architecture is not discussed in this paper, however, it is available in [5, 11–13].

Therefore, the purpose of this systematic review paper is to investigate the different types of sensor data errors which contribute to the degradation of sensor data quality and the existing solutions to detect and correct those errors which can be applied in any layer of the IoT architecture. The different domains the solutions are presented in and the datasets used for evaluation are also studied. This systematic review acts as an introduction for new researchers to the field of sensor data quality or as a guide for researchers who are interested in the techniques used to solve problems related to the sensor data quality topic. In short, a systematic review is a rigorous and structured way of conducting a literature review which allows it to be reproducible. It also helps researchers identify knowledge gaps in the area of interest by extracting and analysing existing solutions. Other review papers about sensor data quality are present, such as the works of Li et al. [14] and Prathiba et al. [15]. However, those review papers do not mention the methods used with respect to the different type of sensor errors and are not systematic reviews.

This systematic review also focuses on stationary wireless sensor networks. This is because many of the mobile sensor network problems are related to network connectivity issues rather than sensor data quality. The field of imaging has also been excluded as it is found that the methods used to improve image data quality varies significantly compared to other physical sensor data. The remainder of this paper is organized as follows: "Research methodology" section describes the methodology used in the systematic review and the results from the review are provided in "Results" section. A discussion about the challenges found in the research area is presented in "Discussion" section. Lastly, "Conclusion" section concludes the study.

Research methodology

A systematic review is a standardized way of extracting and synthesizing information available from existing primary studies with respect to a set of defined research questions. It helps researchers focus on the topic at hand and to identify knowledge gaps in a research area. It is frequently used in the field of medicine and though not as common in the field of computer science [16], a systematic review is still applicable and beneficial in terms of providing a formal way of conducting a computer science-related literature review.

The systematic review in this paper follows the guidelines of Kofod-Petersen [16] and Silva and Neiva [17] for conducting a systematic review in computer science-related fields. It is also done in accordance with the PRISMA [18] (**P**referred **R**eporting **I**tems for **S**ystematic **R**eviews and **M**eta-**A**nalyses) checklist which is an "evidence-based minimum set of items for reporting in systematic reviews and meta-analyses". Since the PRISMA checklist is constructed mostly for medical review literature, some of the items such as the meta-analyses criteria are not considered in this review paper.

The systematic review process is broken down to several steps, starting with the definition of research questions in which this paper aims to answer. Next, the search process and strategy are described, which specifies the keywords and search string used to find the relevant and available publications literature databases. The search strategy also involves a topic modelling step which was carried out to help refine the keywords and search string. The inclusion and exclusion criteria, as well as the quality criteria, are then defined to assist with the selection of relevant literature. Next, data extraction is carried out which extracts data such as the title, abstract and publication year of the literature as well as the types of sensor errors addressed, types of methods for detecting or correcting errors and the domain from the selected studies after screening which is then synthesized and presented in the next section, "Results". Finally, the risk of bias or limitations of this review process is discussed. The steps for the systematic review and its risk of bias are described in detail in the following subsections.

Research questions

The motivation for this systematic review is to provide new researchers an introduction to the field of sensor data quality and the errors that might occur, or as a guide for researchers who are interested in solving sensor data quality related issues. Thus, the following research questions (RQs) are designed in which this paper aims to answer:

- RQ1: What are the different types of errors in sensor data?
- RQ2: How to quantify or detect errors in sensor data?

- RQ3: How to correct the errors in sensor data?
- RQ4: What domains are the different types of methods proposed in?

With RQ1, we are able to investigate the common errors that leads to the degradation of sensor data quality in this field. Moreover, RQ2 allows us to find existing solutions to quantify or detect the aforementioned errors and RQ3 takes it one step further by finding techniques to correct them. RQ4 on the other hand, gives an insight to how various domains use different (or similar) techniques to solve sensor data quality problems and the datasets used to evaluate the methods.

Search process

For this review paper, three computer science-related literature databases are used to search for relevant literature about sensor data quality. The three databases are:

- ACM Digital Library[1]
- IEEE Xplore[2]
- ScienceDirect[3]

These databases are last searched on the September 27th 2018 and the search results are exported into BibTeX format which is then downloaded and stored in the reference manager Zotero. [4] For ACM Digital Library, the export function for BibTeX format only exports the citation data of the literature, but not the abstracts. Thus, Zotero's Google Chrome plugin is used which allows the citation information, including the abstract, to be imported directly into Zotero.

Improving search strategy by topic modelling

At the start of the search process, the keywords defined are "sensor data" and "data quality" which are used in the initial search string:

$$\text{“}sensor\ data\text{''}\ AND\ (quality\ OR\ \text{“}data\ quality\text{''}\ OR\ \text{“}sensor\ data\ quality\text{''}) \tag{1}$$

The initial search results using query (1) returned 13,057 publications from three databases, ACM Digital Library, IEEE Xplore, and ScienceDirect. In order to check, if this initial search query retrieves publications which match the scope of this review, we are using a text mining approach from natural language processing known as topic modelling. Topic models are "probabilistic models for uncovering the underlying semantic structure of a document collection based on a hierarchical Bayesian analysis of the original texts" [19, p. 71]. The idea of the topic modelling step is to identify keywords and groups of keywords that describe the content of the initial set of publications returned by search query (1). In order to do so, topic modelling via Latent Dirichlet Allocation (LDA) [20] is used to find groups of words that are likely to occur together and represent

[1] https://dl.acm.org.

[2] https://ieeexplore.ieee.org/Xplore.

[3] https://www.sciencedirect.com.

[4] https://www.zotero.org.

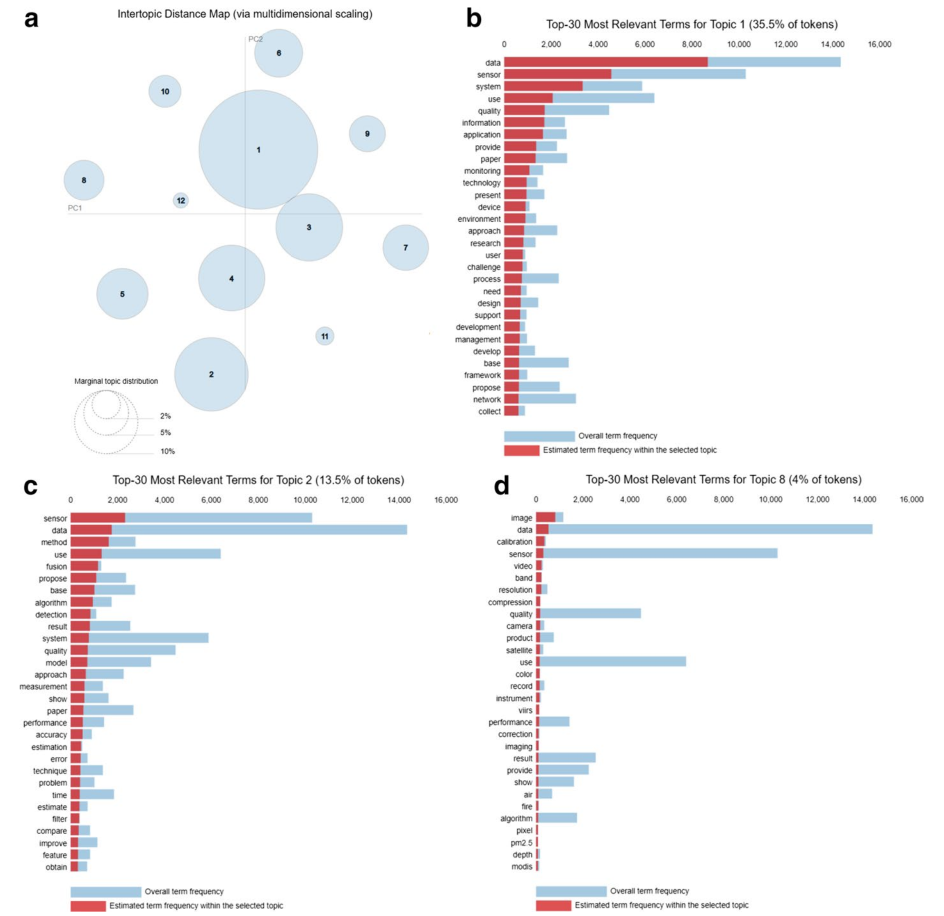

Fig. 2 Topic modelling with LDA. Topic model of 13,057 titles and abstracts found from the search string 1. Topic modelling is performed using a LDA model with 12 topics: **a** intertopic distance showing the marginal topic distribution of each topic, **b**–**d** top 30 most relevant terms for Topic 1, 2 and 8 respectively, along with its estimated term frequencies. Topic 1 and 2 have terms related to sensor and data, but Topic 1 is more focused on systems and applications whereas Topic 2 is associated with methods and algorithms. Topic 8 consists mostly of keywords related to imaging and satellite imaging

a specific topic. For example, assume that the researcher decides to model three topics, named Topic A, B, and C for convenience. After fitting, the LDA model assigns each document the probability of it covering a specific topic, e.g. Document 1 has a 20% probability of being in Topic A, 75% being in Topic B and 5% being in Topic C. The LDA learns these topic models by going through each document and cluster words that have a high likelihood of term co-occurrence. By analysing the words that describe the cluster, the researcher can then interpret the topic for each cluster.

Here, the LDA model is implemented using scikit-learn's [21] estimator `LatentDirichletAllocation`. For the purpose of this analysis, the title and abstract of the publications, which have been identified from search query (1), are used for modelling the underlying topics. The visualization of the LDA model with 12 topics obtained from the 13,057 documents (title and abstracts) of search string (1) is shown in Fig. 2a with the intertopic distance showing the marginal topic distribution. Figure 2b–d lists the top 30 most relevant terms for Topic 1, Topic 2 and Topic

8 respectively. Topic 1 and Topic 2 both have top terms related to sensor and data. However, Topic 1 seems to be more focused on systems and applications, whereas Topic 2 is more related to methods and algorithms. Looking at the top 30 keywords of Topic 8, one might classify that topic as "Imaging" or "Satellite Imaging" since words such as "image", "video", "resolution", "camera", "satellite" and "pixel" occur in that cluster. Through this topic modelling step, it can be seen that there are a handful of papers related to "imaging" in the initial search results. Because imaging is a topic we do not want to focus on, we are using the terms of Topic 8 to refine the search string and set them to be one of the exclusion criteria in this paper.

Through the topic modelling, we decided that the field of imaging is not to be considered in this paper as the techniques used for improving image data quality is very different compared to other physical sensor data. It is made an exclusion criterion (see "Inclusion and exclusion criteria") and the final search string used to search the literature databases is defined as:

$$\begin{aligned}&\textit{sensor data AND } (\textit{quality OR ``data quality" OR ``sensor data quality"})\\&\textit{AND NOT } (\textit{``imaging"}) \textit{ AND NOT } (\textit{``satellite imaging"})\end{aligned} \tag{2}$$

However, readers interested in that field of research can look at review papers [22–24] that investigates data quality in imaging, e.g. camera captured document images and healthcare imaging.

Inclusion and exclusion criteria

The eligibility criteria are criteria used for screening and selecting relevant literature from the search results. The eligibility criteria are composed of the inclusion and exclusion criteria. As mentioned in "Introduction" section, this systematic review focuses on stationary wireless sensor networks as mobile sensor networks tend to lean towards network connectivity issues. Moreover, the field of imaging is to be excluded as the techniques used for improving data quality for images vary significantly from physical sensor data.

Inclusion criteria (IC):

IC1 Papers that involve sensor data,
IC2 Papers that mainly focus on data quality of sensor data,
IC3 Papers about stationary wireless sensor network,
IC4 Papers that consider different types of sensors.

Exclusion criteria (EC):

EC1 Papers that are duplicates,
EC2 Papers not in English,
EC3 Papers without methodology,
EC4 Papers that are secondary studies (e.g. survey, reviews, demos, posters, tutorials),
EC5 Papers about imaging (camera images, 3D images, video streams) or satellite imaging.

Even though imaging-related publications are directly filtered from the search query (2), EC5 is added as an exclusion criterion because there are still publications with imaging-related topics present in the result obtained from the search. This is due to the use of the same search string for all databases, which produces different results in different databases. For example, in the substring *" ...AND NOT ("imaging") ..."* of search string (2), ACM Digital Library removes all lemmatized terms related to *"imaging"* e.g. images, image, imaging but IEEE Xplore removes only the specified word "imaging".

Study quality assessment

In addition to the inclusion and exclusion criteria, the quality criteria are defined to evaluate the quality of papers selected after full-text screening. Using these criteria, the quality of the selected literature can be assessed to see if they are fully appropriate for this systematic review, based on their importance with respect to answering the research questions. The following are the quality criteria (QC):

QC1 Does the study contain validation?
QC2 Does the study propose a way to quantify/detect uncertainty?
QC3 Does it propose a solution to correct the uncertainty/erroneous data?

The papers are scored according to whether they are able to meet the above quality criteria i.e. Yes, No, or Partially. The scores are Yes $= 1$, No $= 0$ and Partially $= 0.5$. It is seen that 54 out of 57 publications have a QC score of two or above and only three publications [25–27] have QC scores of one, which shows that the quality of the majority of the selected literature is of good quality and they are relevant to the systematic review. The three publications with a QC score of one are still included in this systematic review as two of them are related to enterprise-level systems, which gives an insight into how existing methods are integrated into practice. The third paper, which is one of the earliest publications that presented a PCA-based solution for sensor fault detection, is also included as it is highly cited by other papers that proposed PCA-based methods.

Study selection

The initial search query (1) returned 13,057 publications. After the process of topic modelling, the refined search query (2) resulted in 6970 publications. The 6970 publications obtained from the three literature databases are then screened to remove duplicates. There are 107 duplicates which are removed. Next, the duplicate-free set of papers are screened based on their title and abstract. Irrelevant papers, based on the inclusion and exclusion criteria, are excluded and this resulted in the selection of 285 papers. Those screened papers are then read and evaluated in full-text to assess based on their ability and contribution to answering the research questions. About 228 papers are considered irrelevant and are rejected and the other 57 papers that are eligible are chosen to be included in the study.

The selection process is visualized in Fig. 3 as a PRISMA flow diagram [28], showing the number of papers obtained from each stage of the review process i.e. search results, duplicate removal, title and abstract screening, full-text screening, and final selected papers.

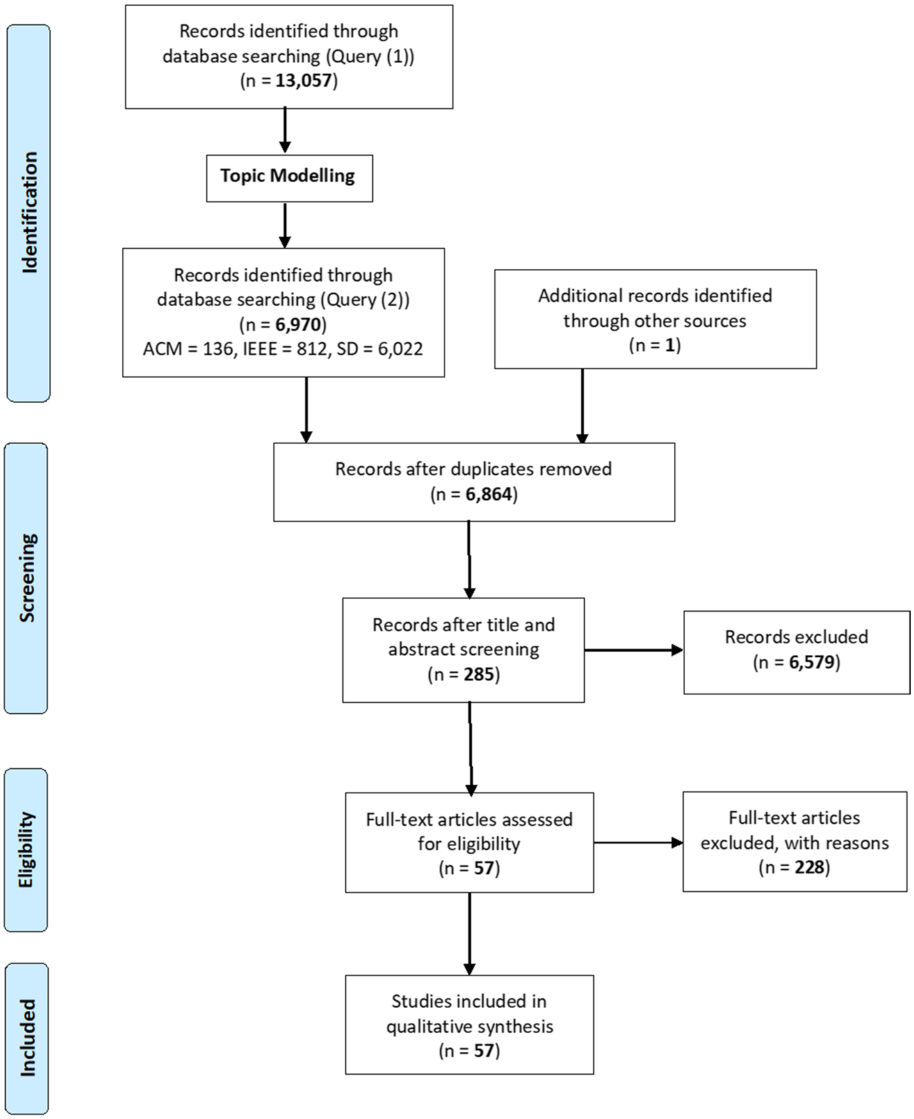
Identification
Screening
Eligibility
Included
Records identified through database searching (Query (1)) (n = **13,057**)
Topic Modelling
Records identified through database searching (Query (2)) (n = **6,970**) ACM = 136, IEEE = 812, SD = 6,022
Additional records identified through other sources (n = **1**)
Records after duplicates removed (n = **6,864**)
Records after title and abstract screening (n = **285**)
Records excluded (n = **6,579**)
Full-text articles assessed for eligibility (n = **57**)
Full-text articles excluded, with reasons (n = **228**)
Studies included in qualitative synthesis (n = **57**)

Fig. 3 PRISMA flow diagram. Extended PRISMA flow diagram visualizing the selection process starting from the 13,057 publications from the initial topic model to the 6970 publications obtained from search query (2) and the subsequent selection process resulting in the final number of 57 considered publications

Data extraction

Data extraction is carried out for all 57 selected publications and the results are tabulated using an Excel spreadsheet. The data extracted from the selected literature are:

- Title and abstract of literature,
- Authors' names,
- Database,
- Publication year,
- Types of sensor data errors addressed (RQ1),
- Types of methods for detecting or mitigating errors (RQ2 and RQ3),
- The domain in which the methods have been developed (RQ4).

Data synthesis

After the data extraction step, the extracted data is analysed to answer the research questions. For RQ1, the definitions of the different types of errors addressed in the papers were analysed as they might have been termed differently in different publications but referred to the same type of error ("Types of errors in sensor data"). Once establishing the definitions of each error, they are then classified so that the errors with the same definition are in the same category. For RQ2 and RQ3, the different types of methods proposed in the literature are analysed and their state-of-the-art techniques are categorized and studied ("Methods for detecting and quantifying errors in sensor data", "Methods for correcting errors in sensor data" and "Methods for detecting and correcting errors in sensor data"). The extracted domains are extracted along with publicly available datasets used for method evaluation in those domains to answer RQ4 ("Types of domains"). Moreover, for literature with validation, the evaluation conditions and results of the methods are also analysed to compare and identify the gaps in knowledge ("Discussion").

Risk of bias

This systematic review is not without bias. Firstly, there is a risk of bias in the review process as only one reviewer screening the literature where the subjectivity of the inclusion and exclusion criteria may affect the selection of relevant publications. Moreover, the year range was not specified during the search process. This means that the search results returned are from all available years, that is from the earliest publication found in the respective databases until recently (September 2018). The databases returned different earliest start years e.g., the earliest publication from ACM Digital Library is from 1998, IEEE Xplore is from 1979, and ScienceDirect is from 1995.

Furthermore, there are publications missed in the search process because the search was done only on three databases, and there are many more databases (e.g., Google Scholar, Scopus, SpringerLink) that might have other literature addressing the mentioned sensor data quality problems. Thus, this systematic review paper is not an exhaustive list of methods available for detecting and correcting sensor data errors. Other than that, there was no snowballing done in this systematic review, i.e. the review process did not include searching and extracting information from the references of the selected papers for the purposes of this systematic review.

Results

This section presents the findings from the extracted data with respect to the research questions formulated in "Research questions" section. In "Types of errors in sensor data" section, RQ1 is addressed to discuss the different types of errors that exist in sensor data which leads to the degradation of sensor data quality. Next, in "Methods for detecting and quantifying errors in sensor data", "Methods for correcting errors in sensor data", and "Methods for detecting and correcting errors in sensor data" sections, RQ2 and RQ3 are answered with respect to the type of errors. The nomenclature in Table 1 is used in those three subsections. "Methods for detecting and quantifying errors in sensor data" section addresses methods proposed only for fault detection and uncertainty quantification (RQ2) and "Methods for correcting errors in sensor data" section discusses solutions

Table 1 Nomenclature used for "Methods for detecting and quantifying errors in sensor data";"Methods for correcting errors in sensor data" and "Methods for detecting and correcting errors in sensor data" sections

Symbol	Description
$x_i(t_j)$	Measured data value x_i of sensor i at a specific point in time t_j
$\hat{x}$	Estimated sensor data value
$\vec{x}$	Sensor data vector, where $\vec{x} = (x_1, \ldots, x_i, \ldots, x_V)$ is a row vector obtained at the same point in time
t	Time in sensor data stream, e.g x_t is the observed sensor data value at time t
i	Column index $i = 1, \ldots, V$
j	Row index $j = 1, \ldots, N$
f	Feature
q	Size of moving window
N	Number of samples
V	Number of variables e.g. temperature, humidity, voltage
M	Number of sensor unit
F	Number of features
$\mathbf{Z}$	Sensor data stream in the form of a time series, $\mathbf{Z} = \left(\ldots, \vec{x}_{t-1}, \vec{x}_t, \vec{x}_{t+1}, \ldots\right)$
$\mathbf{X}$	Sensor data matrix where $\mathbf{X} \in \mathbb{R}^{N\times V}$, $\mathbf{X} = \left(\vec{x}_1, \ldots, \vec{x}_j, \ldots, \vec{x}_N\right)$

Depending on how the samples are obtained, the variables in sensor data vector $\vec{x}$ might be produced by more than one sensor. For example, in environmental monitoring, the data may be produced by several sensors, each measuring one variable e.g. temperature and humidity. On the other hand, some variables are produced by one sensor alone, such as an accelerometer which produces readings for three variables i.e. the acceleration in the direction x, y, and z

for missing data imputation and de-noising (RQ3). As for methods that address both research questions simultaneously i.e. fault detection and correction (RQ2 and RQ3), the results are presented in "Methods for detecting and correcting errors in sensor data" section. This is followed by "Types of domains" section where the domains in which the methods are proposed in (RQ4) are detailed.

Types of errors in sensor data

According to the International Standardization Organization (ISO) [29], an error is defined as "the result of a measurement minus the true value of the measurand". There are several types of errors related to sensor data quality. Table 2 shows the different types of errors extracted from the selected literature (RQ1), along with the papers that address

Table 2 Types of errors addressed, along with its respective papers and total number of papers that address that error

Type of error	Papers	Total
Outliers	[7, 30–60]	32
Missing data	[7, 9, 25, 26, 31, 38, 46, 51, 61–68]	16
Bias	[30–32, 41, 43, 59, 60, 69–73]	12
Drift	[31, 32, 34, 35, 54, 60, 69, 70, 72–75]	12
Noise	[35, 52, 53, 72, 73, 75–77]	8
Constant value	[30, 35, 52, 53, 72, 73, 78]	7
Uncertainty	[25, 26, 68, 79–81]	6
Stuck-at-zero	[30, 32, 53, 72, 73, 78]	6

them and the total number of papers. Note that some literature address different types of errors in the same paper, for example, [30–32] addressed both outliers and bias in their proposed solution.

The type of error that is most commonly addressed in publications related to sensor data quality is outliers and is addressed by 32 papers, which is more than half of the total number of selected studies. *Outliers*, also known as anomalies [82] and spikes [36, 83], are values that exceed thresholds or largely deviate from the normal behaviour provided by the model. A sensor data measurement is also considered an outlier if it is significantly different from its previous and next observations or observations from neighbouring sensor nodes [38, 45, 48]. Outliers are also known as *faults*, though faults also include other types of errors such as bias, drifts, noise, constant value, and stuck-at-zero. Though some papers [50, 55] might not have specified the type of fault, most of them breakdown the fault error to the different types of errors as mentioned previously.

The second most commonly found error in sensor data is *missing data*, which is addressed in 16 publications. It is also known as incomplete data, and it is one of the data quality (DQ) dimensions introduced by Wang and Strong [4]. DQ dimensions categorize the quality of data in databases that are important to data consumers. They are mostly used to describe data in enterprise-level systems and are used for modelling how data errors propagate to the consumer's end. However, apart from incomplete (missing data) and inaccurate data (uncertainty), which are sensor data quality-related issues, other DQ dimensions such as inconsistent data and timeliness are not considered in this review paper as they are more related to the topics of database design or communication data quality. According to Li and Parker [9], missing data is caused by various factors such as unstable wireless connection due to network congestion, sensor device outages due to its limited battery life, environmental interferences e.g. human blockage, walls, and weather conditions, and malicious attacks. There are cases where sensor data is missing for extended periods of time, which might lead to incorrect decision making on the consumer side. Though the simplest way to solve this problem is to re-transmit the data, most IoT applications are in real-time, which would render the data useless if there is a delay. Besides that, the computational and energy cost causes it to be inefficient as these sensor devices are usually limited in terms of battery, memory, and computational resources.

Bias, also known as an offset, is a fault with a constant offset or as Rabatel et al. [84] defines, "a value that is shifted in comparison with the normal behaviour of a sensor". This type of error would usually require calibration to subtract the offset from the observed reading to get its true value. *Drifts* are readings that deviate from its true value over time due to the degradation of sensing material which is an irreversible chemical reaction [60] whereas *constant values* are readings with a constant value over time, though it might belong to a normal range. It is usually caused by a faulty sensor or transmission problems [84]. Another type of fault is a *stuck-at-zero* or dead sensor fault. As its name implies, it refers to values that are constantly at zero over an extended period of time. Lastly, *noise* is also a type of fault, and they are small variations in the dataset. Noise is similar to *uncertainty*, which is another type of error and DQ dimension. According to the ISO [29], the definition of uncertainty is, "a parameter, associated with the result of a measurement, that characterizes the dispersion of the values that could

Table 3 Types of methods addressing error detection and quantification (RQ2) only, along with its addressed errors, respective papers, and the total number of papers that proposed that method

Method	Errors addressed	Papers	Total
Principal component analysis	Outliers, bias, drift, stuck-at-zero	[27, 32, 47, 48, 50, 59, 69]	7
Artificial neural network	Outliers, bias, drift, constant values, noise, stuck-at-zero, uncertainty	[34, 36, 54, 70, 78, 81]	6
Ensemble classifiers	Outliers, drift, constant values, noise, uncertainty	[33, 35, 37, 79]	4
Support vector machine	Outliers	[57, 58]	2
Clustering	Outliers	[39, 45]	2
Ontology/knowledge-based systems	Uncertainty (inaccurate data), missing data (incomplete data)	[25, 26]	2
Univariate autoregressive models	Outliers	[40]	1
Statistical generative models	Outliers	[49]	1
Grey prediction model	Outliers, noise, constant values	[52]	1
Particle filtering	Bias, scaling	[71]	1
Association rule mining	Outliers	[56]	1
Bayesian network	Outliers, noise	[44]	1
Euclidean distance	Outliers	[42]	1
Hybrid methods			
Polynomial predictive filter and fuzzy rules	Outliers	[53]	1
Dempster–Shafer theory and mathematical modelling	Drift, noise	[75]	1

reasonably be attributed to the measurement". Thus, uncertainty can also be seen as the quantification of an error in statistical terms. Moreover, Mansouri et al. [47] states that sensor data uncertainty includes "measurement noise, sensor imprecision and variability of measured quantity". According to that definition, noise contributes to uncertainty. However, the methods of correcting those two errors are relatively different where noise correction techniques mostly includes signal processing solutions whereas uncertainty quantification and correction involves ontology-based methods (refer to "Methods for detecting and quantifying errors in sensor data", "Methods for correcting errors in sensor data" and "Methods for detecting and correcting errors in sensor data" sections).

Methods for detecting and quantifying errors in sensor data

Most solutions suggest ways to quantify or detect errors in existing literature either address the *detection of faults* i.e. outliers, bias, drifts, constant values, or to *quantify the uncertainty* in the sensor data. These publications only address the problem of detecting those errors, but not correcting them. There are 32 publications that proposed methods to solve this problem, which is 56% of the total number of selected literature. Table 3 obtained from the data extraction process of the 32 papers shows the different existing methods to quantify or detect sensor data errors (RQ2), along with the errors addressed, the respective papers that presented the method and the total number of papers. It can be seen that the three most common approaches are principal component analysis, artificial neural network and Ensemble Classifiers, which constitutes more than half of the reviewed publications which proposed error detection and quantification methods, with 7, 6 and 4 papers proposing those methods respectively. There are also hybrid approaches, which incorporates more than one type of method in detecting sensor data errors. The following is a brief overview of each method, where "Anomaly/fault detection" section discusses methods for detecting anomalies or faults in the sensor data and "Uncertainty quantification" section presents approaches for quantifying the quality of the data.

Anomaly/fault detection

Firstly, to detect faults, several methods such as statistical and machine learning, clustering, ontology, and hybrid approaches have been suggested.

Principal component analysis (PCA) Principal component analysis (PCA) [27] is commonly used to find patterns in the data i.e. the correlation between variables, by generating orthogonal principal components. Therefore, other than being used as a feature reduction technique, PCA can also be used for fault detection. In sensor data matrix $\mathbf{X}$ with N rows (measurements at different points in time) and V columns (measurement of different sensors), PCA is done by firstly standardizing the matrix $\mathbf{X}$ if the variables are of different units of measurements (e.g. ^{o}C, lux, km/h) or if each variable is to receive equal weight in the analysis. To standardize the matrix, each data point $x_{j,i}$ of matrix $\mathbf{X}$ is subtracted by the mean of the respective column μ_i and the differences $(x_{j,i} - \mu_i)$ are divided by the column's standard deviation σ_i. This process is known as whitening in statistics. Next, the covariance matrix $\mathbf{X}^T\mathbf{X}$, which quantifies the correlation between each of the variables, is calculated by multiplying the transpose of the standardized sensor data matrix with itself. The eigenvectors and their corresponding eigenvalues of the covariance matrix are calculated [85, Chap 11], which produces two matrices: $\mathbf{P}$, which is the modal matrix where the columns are eigenvectors and $\mathbf{D} = \mathbf{P}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{P}$, which is the spectral matrix where the diagonal elements are the eigenvalues. The pairs of eigenvalues and eigenvectors are sorted from largest to smallest eigenvalue, such that the first eigenvector (or principal component) accounts for the largest amount of variance, which is given by the corresponding eigenvalue. The orthogonal transformation

$$\mathbf{T} = \mathbf{X}\mathbf{P}$$

converts the sensor data matrix $\mathbf{X}$ into a set of values from linearly uncorrelated variables, the so-called principal components. The top few principal components capture most of the variability in the dataset. This is also how it is used as a dimension-reduction technique, because it projects the dataset into a lower-dimensional subspace.

Following the steps for PCA, two orthogonal projection subspaces are obtained from the selection of the top few principal components and the standardized data matrix $\mathbf{X}$ can then be decomposed into the following:

$$\mathbf{X} = \hat{\mathbf{X}} + \mathbf{E}\,,$$

where $\hat{\mathbf{X}}$ is the principal component subspace which is the modelled variations of $\mathbf{X}$ and includes the signal in the dataset, containing the first l eigenvectors i.e. the first l columns of $\mathbf{P}$ (where l is the number of selected principal components) and $\mathbf{E}$ is the unmodelled variations of $\mathbf{X}$, also known as the residual matrix which includes mainly noise and useless information and consist of the last $V - l$ columns of $\mathbf{P}$. They are represented as the following:

$$\hat{\mathbf{X}} = \mathbf{T}\mathbf{P}_\ell^T = \sum_{i=1}^{l} t_i p_i^T = \mathbf{X}\mathbf{C}\,,$$

where $\mathbf{C} = \mathbf{P}\mathbf{P}^T$ and similarly,

$$\mathbf{E} = \mathbf{T}_e\mathbf{P}_e^T = \sum_{i=l+1}^{V} t_i p_i^T\,.$$

Therefore, a new sample vector, $\vec{x}$ can be projected into the principal component subspace:

$$\vec{x} = \hat{\vec{x}} + \vec{e}$$
$$\hat{\vec{x}} = \vec{x}\mathbf{P}\mathbf{P}^T = \vec{x}\mathbf{C}$$

and into the residual subspace:

$$\vec{e} = \vec{x} - \hat{\vec{x}}\,,$$
$$\vec{e} = \vec{x} - \vec{x}\mathbf{P}\mathbf{P}^T = \vec{x}(\mathbf{I} - \mathbf{C}).$$

$\mathbf{C}$ and $(\mathbf{I} - \mathbf{C})$ are also known as the model projection matrix and residual projection matrix respectively. Fault detection can be done by monitoring the residual subspace of the PCA as it increases in magnitude when there is a change in the correlation among the variables in x. The squared prediction error, also known as Q-statistic, defined as:

$$Q = ||\vec{e}||^2 < Q_a\,,$$

where Q_a is the Q-statistic threshold. Details on how to obtain Q_a is found in [32, 42, 69].

Dunia et al. [27] proposed a Sensor Validation Index (SVI) as a means for fault detection and isolation (identifying faulty sensors) through an iterative reconstruction process that assumes each sensor fails, reconstructs the faulty sensor and compare the Q-statistics before and after reconstruction. The SVI, which ranges from 0 to 1, shows that when a sensor is faulty, it is close to zero and vice versa. Alawi et al. [69], on the other hand, introduced a combined contributions index using the Q-statistics, which measures the variance of random noise in the residual subspace and Hotelling's T^2-statistics, which represents the variance in the model subspace. This is because an occurrence of a fault (bias and constant value mentioned in this paper) usually leads to changes in either statistical metric.

Rassam et al. [48] proposed a variation of PCA called the One-Class Principal Component Classifier for local and unsupervised anomaly detection. The approach is divided into two parts, with the first being the offline training phase which trains a PCA model using normal data collected from each sensor to build the normal behaviour model and it is stored locally in each sensor node. The dissimilarity measure is calculated using the sum of squares of the normalized principal components, and this represents the maximum and minimum thresholds for anomaly detection. The second phase is the online detection phase where current observations would be projected into the feature subspace and compared with normal behaviour model based on its the dissimilarity matrix. The normal PCA model is also updated and retrained with new mean and standard deviation of the new data. In order to deal with non-linear systems, Sharifi and Langari [50], suggested a Mixture Probabilistic PCA model for fault diagnosis which separates the input space into several local linear regions and subsequently has linear sensor fault diagnosis applied to each linear region.

Moreover, Zhao and Fu [59] have also proposed a sensor fault detection for outliers and bias using PCA by modelling the normal behaviour for continuous glucose monitoring applications. Harkat et al. [32] also applied the PCA technique to detect outliers, drifts, bias, stuck-at-zero faults. However, rather than just using the SVI [27], a test on the sum of squares of the residual matrix, i.e. the last $(V - l)$ principal components is done to detect faults. Recently, Mansouri et al. [47] came up with another variation of PCA called the Midpoint-radii PCA for fault detection. The Midpoint-radii PCA allows interval-valued data, which considers the uncertainty in the data, to be modelled.

PCA is a powerful technique used for many applications, including fault detection in which 7 out of 32 methods proposed are based on. It can be adapted to multiple variations, which have their own advantages such as the One-Class PCA classifier, which is able to perform locally with no extra communication overhead, making it suitable for Edge Computing applications. However, PCA requires fault-free training data which is rare and difficult to obtain. There is also a need to choose the optimal number of principal components, which differs from one application to another.

Artificial neural network An artificial neural network (ANN) is a framework that is vaguely modelled upon the biological neural network of a brain. It is mainly used to learn patterns or models from complex processes such as pattern recognition. ANNs consist of a densely interconnected set of neurons, also known as perceptrons, whereby each unit takes several real-valued inputs (combined using an input function), runs it through its activation function e.g. linear, sigmoid and rectified linear unit, and produces a single real-valued output. Each input has a weight related to it, which determines the contribution of the inputs to the output. Learning the weights of the input values such that it produces the correct output value is the basis that trains an ANN to learn. There are also many ways of doing so, such as the perceptron rule for linearly-separable datasets, gradient descent for non-linear datasets and backpropagation.

Jäger et al. [78] introduced a framework to detect four different types of fault: outliers, offset, noise and stuck-at-zero, using a supervised time-delay neural network (TDNN). It is a type of multi-layer feed-forward ANN that allows the mapping between past and present values by analysing the sliding windows of a signal. The difference between TDNN and the classic multilayer perceptron is that the neurons receive not only the output from the neurons below but also the delayed (past) outputs of those neurons. However, it is seen that TDNN is only able to detect 2 out of the 4 fault types reliably, namely the offset and stuck-at-zero. Bosman et al. [36] proposed a decentralized learning approach for fault detection i.e. for anomalies such as outliers, drift, noise, and constant values, which learns the normal sensor behaviour model in each sensor node, while

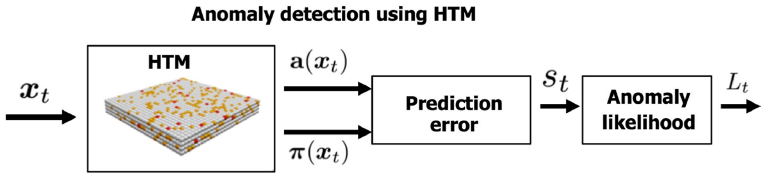

Fig. 4 Framework of HTM for anomaly detection [34, Fig 3(a)] Input vector $\vec{x}_t$ of the data stream is sent to the HTM where it produces a sparse binary vector representing the current input, $a(\vec{x}_t)$ and prediction for the next time step, $\vec{\pi}(\vec{x}_t)$. The prediction error, s_t is calculated and is used to obtain the anomaly likelihood, L_t. *Image obtained under the Creative Commons license. No revisions were made to the image.*

incorporating neighbourhood information. The approach uses Recursive Least Squares to learn linear models and so-called Extreme Learning Machines (see "Artificial neural network" section) for learning non-linear models.

Smarsly and Law [70] suggested a decentralized fault detection and isolation software package framework for bias and drifts using backpropagation feedforward neural network, which is embedded in each wireless sensor nodes of the system. Once again, the Neural Network learns the normal behaviour model of the system and outputs an estimated value in which the current observed value will be compared against and detected if it is anomalous. Xiao et al. [54], on the other hand, introduced an Auto-associative Neural Network (AANN) solution for fault detection and prognosis for outliers and drifts. AANN is a feedforward neural network with an odd number of hidden layers that are used to produce an approximation of the identity mapping between input and output layers (auto-encoder) [86]. It has a bottleneck hidden layer which compresses information, which forces it to eliminate redundancy and capture the input patterns. The faults are detected using shallow and deep AANN, and the prognosis is done using Autoregressive Moving Average.

Ahmad et al. [34] proposed a framework for anomaly detection using hierarchical temporal memory (HTM), a type of unsupervised artificial intelligence learning method based on neuroscience research. It is similar to an artificial neural network, but unlike most neural networks, HTM can learn time-based patterns in an unlabeled data stream. It is firstly described in the book "On Intelligence" by Hawkins and Blakeslee [87] in 2004 and has since been continuously developed by his company, Numenta [88]. Numenta also provides an open-source anomaly detection benchmark, numenta anomaly benchmark (NAB) for evaluating anomaly detection algorithms in real-world streaming data which consists of labelled anomalies. The steps of a HTM is seen in Fig. 4, where the input of the data stream, $\vec{x}_t$ which is an observed data vector at time t is sent to the HTM system. Then, the HTM returns a sparse binary vector representing the current input, $\vec{a}(\vec{x}_t)$ and the prediction for the next time step, $\vec{\pi}(\vec{x}_t)$, which is the estimation of $\vec{a}(\vec{x}_{t+1})$ in sparse binary vector form. The prediction error, s_t is calculated and the probabilistic model of it is used to compute the likelihood of the data being an anomaly, L_t.

Along with PCA, ANN is also another common technique for fault detection and it also has multiple variations such as TDNN, AANN and HTM. There are 6 out of 32 papers that have presented an ANN-based approach, which has its own pros and cons. The advantages and disadvantages depend heavily on the type of Neural Network applied. For example, TDNN has several disadvantages such as not being able to detect noise and outliers reliably and requires many parameter decisions, whereas AANN is robust against training data with missing values.

Ensemble classifiers Ensemble learning use multiple machine learning classifiers to arrive at a better predictive performance compared to when using those algorithms individually by aggregating the results of the classifiers. Bosman et al. [35] proposed a decentralized, online fault detection for anomalies, drifts, noise and constant value using ensemble classifiers where each classifier will learn a normal behaviour model and compare it with the current reading to identify if it is an anomaly. The results are then aggregated using simple heuristic rules or applying algebraic combiners e.g. median or Fisher's method [89]. The classifiers mentioned in the paper are Sliding Window Mean, Recursive Least Squares, Extreme Learning Machines, Polynomial Function Approximation. Curiac and Volosencu [37] also suggested an anomaly detection technique using ensemble-based classifiers which models the normal behaviour of the sensors whose votes (if a sensor reading is anomalous or not as compared with the normal behaviour model) are then collected and aggregated. The types of classifiers used in the paper are the Average-based classifier, Auto-Regressive Linear Predictor-based classifier, Neural Network-based classifier, Neural Network Auto-Regressive Predictor-based classifier and the Adaptive Neuro-Fuzzy Inference System-based classifier.

Abuaitah and Wang [33] introduced a distributed anomaly detection framework to detect anomalies using feature extraction and classification algorithms. The feature extraction is carried out on the child nodes, which incrementally learns new statistical summaries. The features proposed that can be useful to detect anomalies are mean, variance, rate of change, spatial distance, temporal and spatial correlations. The statistical summaries are then sent to the base station (parent node) instead of raw data. There, a classification algorithm such as AdaBoost, Support Vector Machines or simple decision trees is applied to the set of feature vectors received from child nodes. The study showed that AdaBoost performs the best (lowest false positives and negatives) for anomaly detection in their case study. Adaboost converts a collection of weak classifiers (error rate slightly better than random guessing) into a strong one by the weighted combination of the weak classifiers. During classification, the child node is labelled as "normal" or "misbehaving", and the parent nodes will stop using data from "misbehaving" child nodes.

Ensemble classifier is a supervised method and though it mostly achieves better predictive performance than its individual classifiers, it is a complex task to build an ensemble classifier. This is due to the need to choose suitable base classifiers, which may be difficult and complicated, depending on the type of application. Also, based on the individual classifiers chosen, some may require feature extraction and fault-free training examples. Large datasets are also needed to train the supervised classifiers.

Support vector machine A support vector machine (SVM) is a machine learning algorithm that aims to find a hyperplane to separate and classify the data points in an F-dimensional space, where F is the number of features. The features are obtained either directly as the variables themselves, or via a process called feature engineering, which produces new features based on the data and its set of variables. The hyperplane (a line in

2D, a plane in 3D, and so on) is a decision boundary in which data points on one side of the hyperplane belong to one class, and data points on the other side belong to another class. The objective is to find a hyperplane that has the widest margin, i.e. the maximum distance between the two data points from the two different classes. Support vectors are data points that are close to the hyperplane and they are the data points that determine the hyperplane by maximizing the margin of the classifier. For anomaly detection, the decision boundary or hyperplane of the normal data is found such that it encompasses most of the data in the feature space. Then, newly observed data that fall out of the boundary are classified as outliers.

In 2009, Zhang et al. [57] proposed an online outlier detection technique using One-Class (unsupervised) Centered Quarter-Sphere SVM which updates the normal behaviour model of the sensed data based on three time windows. The quadratic optimization problem of modelling the SVM is converted into a linear optimization problem by fixing the center of the mapped data at the origin in the feature space. Here, the data vectors $\vec{x}$ in sensor data matrix $\mathbf{X}$ is mapped into a feature space using a non-linear mapping function such as PCA, which returns the top few principal components that can be used as features. Other than that, a Python package for feature engineering and selection, `tsfresh` [90] can also be used to obtain time series features. The normal behaviour at each time window is learned using One-Class Centered Quarter-Sphere SVM to find the minimum radius (hyperplane), which helps detect *temporal* anomalies. Then, the radius is broadcasted to all spatially neighbouring nodes i.e. sensor nodes that are within communication range, and the median radius is calculated. The online characteristic allows the data can be checked against other neighbouring nodes to identify if the temporal anomaly is also *spatially* anomalous, thus confirming the detection of an actual anomaly.

In 2013, Zhang et al. [58] presented another type of SVM called the One-Class Centered Hyper-Ellipsoidal SVM for anomaly detection. The difference between the Quarter-Sphere and Hyper-Ellipsoidal SVM is that the former uses Euclidean distance as a distance measure, whereas the latter uses the Mahalanobis distance to model the SVM. Those two types of distance measures are commonly used to measure the similarity of the data points. However, the Euclidean distance does not take into account the correlation between variables and only calculates the distance in terms of individual variables. On the other hand, the Mahalanobis distance takes into account the correlation between variables and calculates the distance by combining all variables together, forming a covariance matrix. It is also scale-insensitive, but it comes with a higher computational complexity compared to Euclidean distance. The two variations of SVM are unsupervised, adaptive and distributed.

Clustering Clustering is an unsupervised technique for fault detection which has the advantage of not requiring prior knowledge of the system model or underlying data distribution. However, the optimal number of clusters or cluster width has to be determined by the user. One of the clustering-based outlier detection technique is proposed by Fawzy et al. [39] for WSNs. The algorithm uses an in-network fixed-width clustering algorithm along with nearest neighbor and timestamps which helps to identify if it is an erroneous data or an actual event. It consists of a few steps, starting with pre-processing, where the fixed-width clustering algorithm is applied to the dataset to separate and group the data.

In the fixed-width clustering algorithm, each data point is assigned to a cluster and the data point is within a pre-defined distance from the cluster's center. If there is no such cluster, then a new cluster is created with that data point being its center. Next, the outlier detection step labels each cluster formed as "normal" or "outlier". This is done by calculating the Euclidean distance between one cluster to the other clusters. A cluster is detected as an outlier if its average inter-cluster distance is more than one standard deviation away from the mean inter-cluster distance. The data points in the outlier clusters are then further examined by looking at the neighbouring nodes and timestamps to see if those data points are events or actual anomalies.

Liu et al. [45] presented another example of the clustering method used for outlier detection using Time-Relevant *k*-Means clustering for electric power sensor data. The *k*-means clustering algorithm is used to form initial clusters. The *k*-means algorithm can be done in the following steps, for an input *k*, which is the user-defined number of clusters and a dataset, $\mathbf{X} = \{\vec{x_1}, \vec{x_2}, \ldots, \vec{x_N}\}$ where N is the number of samples:

1. Set centroids (centers of clusters), $c_1, c_2, \ldots, c_k$ at random locations.
2. Repeat until convergence:

   (a) For each sample $\vec{x_j}$, assign the sample to the cluster, s with the nearest centroid, c_s:

   $$\arg\min_s D(\vec{x_j}, c_s) \ ,$$

   where D is the distance function.

   (b) Update the centroids of each cluster c_s, where $s = 1, \ldots, k$ after adding the new sample in the cluster:

   $$c_s = \frac{1}{n_s} \sum_{j=1}^{n_s} \vec{x_j} \ ,$$

   where n_s is the number of points in that cluster s.
3. Stop when none of the cluster assignments change, i.e. converge.

In order to choose the appropriate k number of clusters, the quality of the clusters is measured by the Mean Index Adequacy, which calculates the average distance between the cluster center and all the other data points in that cluster. The smaller the Mean Index Adequacy, the better the clustering results. After performing *k*-means clustering using the appropriate number of clusters, the data within each cluster are re-clustered according to the temporal attribute of the data. Outliers are then detected by comparing the current value with the minimum and maximum data value from each refined cluster. An outlier correction method is also considered in that framework, though it is by simple statistical approaches such as imputing the erroneous data using the mean, median and mode values.

Univariate autoregressive models A univariate autoregressive model is a time series model which, using sensor measurements from the previous time step in a moving window, $Z = \{x_{t-q+1}, \ldots, x_t\}$ as input, predicts the value at the next time step, $\hat{x}_{t+1}$. Hill and Minsker [40] proposed an anomaly detection technique using univariate autoregressive models to model environmental data streams. The different models used and compared are the nearest cluster, single-layer linear network, and multilayer perceptron. The nearest cluster estimates the next value as the average of k most similar (based on Euclidean distance) sensor measurements in the dataset, whereas the single-layer linear network predicts the next value based on the linear combination of the q previous measurements. After the predictive modelling, the next sensor data observation can be classified as anomalous by comparing it with the threshold calculated by the prediction interval value. Though it is found that the multilayer perceptron works best in their case study, it might not be the case for other applications.

Statistical generative models Statistical generative models are probabilistic models that attempt to describe how data is generated by learning the statistical distribution of the dataset. For anomaly detection, Sallans et al. [49] presented a statistical generative modelling technique in which new observations will be compared against, and if that new observation has a low probability in that model, then it is counted as anomalous. Examples of statistical generative models used in the paper are the Gaussian model, Hidden Markov model, and Histogram.

Grey prediction model Grey systems theory, initially proposed by Deng [91] in 1982, is developed to cope with the uncertainty of a system and has the advantage of being able to model a discrete time series with a small sample size. It does not require prior knowledge of the underlying data distribution and requires only a small set of training data. In grey systems, some part of the information is known and some part is unknown, thus having incomplete information. The subsequence of the original time series data **Z** helps predict the future value and can be defined as:

$$\mathbf{Z}^{(0)} = \{x^{(0)}(u) + c\}, u = t - q + 1, t - q + 2, \ldots, t; t \geq q; q \geq 3\ ,$$

where $\mathbf{Z}^{(0)}$ consist of the q subsequent observed values up to time t and c is a constant that satisfies $x^{(0)}(u) + c \geq 0$.

The original subsequence is firstly smoothed by an accumulate generating operation (AGO). The first-order AGO is defined as:

$$\mathbf{Z}^{(1)} = \{x^{(1)}(u)\} = \left\{ \sum_{i=t-q+1}^{u} x^{(0)}(i) \right\}, u = t - q + 1, t - q + 2, \ldots, t; t \geq q.$$

The data series obtained after AGO smoothing can be modelled by a simple first-order differential equation to give a grey system model GM(1,1). The grey differential equation is as follows:

$$\frac{dx^{(1)}(u)}{du} + az^{(1)}(u) = b$$

where a and b are parameters and $z^{(1)}(u)$ is the adjacent mean generating operation. The papers [30, 52, 60] provide detailed explanation on how to derive the differential equation. Tsang [52] introduced a sensor data validation technique involving outliers, noise and constant values using grey models where sensor values are compared to the predicted value of the grey model. The parameters, a and b of the GM(1,1) model is estimated using the recursive orthogonal least-squares estimation algorithm.

Particle filtering Particle filtering is a state estimation technique given partial and noisy observations in a dynamic system. It is a Monte Carlo algorithm which uses a set of samples called particles, to represent the posterior probability distribution of a stochastic process. Essentially, the samples from the distribution are rendered as particles and each particle has a weight assigned to it that represents the probability of drawing that particle such that it is close to the actual observed value. It is thus able to model non-linear or non-Gaussian data.

Tadić and Đurović [71] proposed a sensor fault diagnosis technique for bias and scaling errors using particle filtering. Particle filtering is used to estimate the states of the non-linear model, and new observations are compared with the estimated particle to detect whether it is a calibration (bias and scaling) fault. This is done by calculating the residuals, which is the difference between the particle filter's estimate and the current observed data and a fault is detected if it is more than a user-specified threshold, since the residuals are expected to stay close to zero.

Association rule mining Association rule mining is a rule-based machine learning algorithm which can be used for error detection and also missing data imputation (see subsection "Methods for correcting errors in sensor data"). Association rule mining detects frequent patterns, correlations, or causal structures by revealing how items are associated with each other. It helps in predicting the occurrence of a specific item based on the occurrence of other items and is traditionally used for transactional items e.g. product placements in supermarkets. It comprises of the antecedent which is something that is found in the dataset, A, and the consequent, B, which is something that is found in combination with the antecedent. In time series analysis, an association rule $A \implies B$ means that if event A occurs somewhere in the dataset, it will most likely be followed by B. However, to use association rule mining in time series analysis, the data has to firstly be discretized into a pattern e.g. a string of symbols.

There are many different ways to measure association and the most used ones are support and confidence. Support is the measure of how frequent an itemset (or an event followed by another event) is in the dataset whereas the confidence of a rule is the measure of how likely an event A occurs when event B occurs. For time-series analysis, the support of a rule is calculated by:

$$sup(A \implies B) = \frac{Count\ of\ A\ followed\ by\ B\ occuring}{(k - |AB| + 1)},$$

where k is the length of the discretized pattern and $|AB|$ is the length of the pattern AB (A followed by B). The confidence of a rule is:

$$conf(A \implies B) = \frac{sup(A \implies B)}{sup(A)},$$

which tells us the number of times the relationship is found to be true.

Yu et al. [56] presented an Apriori Association Rule Mining method to improve data quality by detecting anomalies (unusual change in time series patterns) in soil moisture probes. The events are discretized and Dynamic Time Warping is used firstly to align and compare the events of different lengths. The Apriori algorithm is a method that reduces the computational complexity of finding rules that are above the support and confidence thresholds (strong rules) by reducing the number of candidate itemsets. The Apriori principle states that if an itemset is frequent, then all of its subsets must also be frequent, and vice versa. By comparing the current observed event to historical records via association rules, anomalies are detected.

Bayesian network A Bayesian network, also known as a belief network, is a probabilistic graphical model that uses a directed acyclic graph to model a set of variables and their conditional dependencies based on Bayesian inference. It can be used to obtain the posterior probabilities of an unknown variable given evidence from other measured variables. The joint probability distribution of the variables, A, B, C, and D is represented as, according to the Chain Rule of probability:

$$P(A,B,C,D) = P(A) * P(B|A) * P(C|B,A) * P(D|C,B,A).$$

It also follows the Local Markov property, which states that each variable is conditionally independent of its non-descendants given its parent variables, which simplifies the Chain Rule into a simpler form.

Ibarguengoytia et al. [44] proposed a Bayesian network approach for detecting and isolating faults e.g. outliers in sensor networks for a gas turbine using two Bayesian networks, one for validation and another one for isolation. For validation i.e. detection of faults, the fitted Bayesian network model is used to produce an estimate. This is done by taking the particular sensor as a hypothesis while the other related sensors act as the evidence. The output, which is the posterior probability distribution of the specific variable, is used to estimate the probability of measuring the recorded sensor data value. If the probability is less than a user-defined threshold, then it is identified as anomalous. In this case, another Bayesian network is created to isolate the fault i.e. to evaluate if it is an event or an actual anomaly. When a faulty sensor actually exists, the fault will be manifested in all the related variables. This can be detected in its Markov blanket, which is the set of variables that makes the variable independent from the others, such as the parents, children, and spouses of the variable. However, the downside to Bayesian Networks is that it requires expert knowledge to form the probabilistic model of the relations between the variables.

Euclidean distance For systems which use PCA for fault detection, Hu et al. [42] proposed a data-cleaning solution using an Euclidean distance approach. The data-cleaning solution aims to remove outliers in the training data as they can strongly affect the covariance structure of the PCA method, which in turn affects the performance of the PCA-based fault detection. This can be done by calculating the z-score of the Euclidean distances of the samples, which converts the multivariate problem into a univariate data comparison. After standardizing the original data matrix, **X**, the training data is now a $N \times V$ standardized matrix, with N being the number of training samples and V the number of variables. The Euclidean distance of the jth row, D_j is defined as:

$$D_j = \sqrt{\sum_{i=1}^{V}(x_{j,i})^2}$$

where $x_{j,i}$ is the ith variable of the jth sample in the normalized data matrix. The mean of the Euclidean distance of all samples, μ_D is:

$$\mu_D = \frac{1}{N}\sum_{j=1}^{N} D_j,$$

and the standard deviation of all samples, σ_D is:

$$\sigma_D = \sqrt{\frac{1}{(N-1)}\sum_{j=1}^{N}(D_j - \mu_D)^2}.$$

The z-score, which is used to identify outliers in the dataset, is calculated as:

$$z_j = \frac{|D_j - \mu_D|}{\sigma_D}.$$

If the z-score of the Euclidean distance of a sample is more than two standard deviations away from the mean, then it is classified as an outlier and is removed.

Hybrid methods Tsang and Chan [53] came up with a sensor validation technique using *predictive polynomial filters* to model the behaviour of normal sensor data and *fuzzy rules* to detect faults such as outliers, random error and sensor failure from the error sequence generated from the model. Predictive polynomial filters divide the signal into small segments and the small segments are modelled by low degree polynomials. Another hybrid approach for fault detection uses *mathematical modelling* and *Dempster–Shafer Theory*, proposed by Zahedi et al. [75]. It is an online approach for detection drifts and noise, which consist of local and global tiers. For local tiers, fault analysis is done and fault vectors are generated by First Order Linear model. For global tiers, fault analysis is done by refining the result from local tiers using the spatial correlation between sensors, Dempster–Shafer theory for sensor fusion which uses the faulty behaviour information to generate a robust estimate of the event of interest. The generated reference signal (ground truth) is fed back to the local tier.

Uncertainty quantification

In order to quantify the uncertainty in the sensor data, the following approaches have been introduced.

Artificial neural network For the purpose of uncertainty quantification, Wang et al. [81] used a special type of learning algorithm for artificial neural networks called Extreme Learning Machines (ELM). This term refers to a new learning algorithm for single hidden layer feedforward neural networks (SLFNs) proposed by Guang-Bin Huang et al. [92], which randomly assigns input weights and analytically determines the output weights of SLFNs. For an SLFN, the output function of the kth hidden node, h_k is

$$h_k(x_j) = G(w_k, b_k, x_j),$$

where w_k and b_k are the parameters, i.e. the weight and the bias or impact factor of the kth hidden node from the input node. The activation function, G is a non-linear piecewise continuous function such as the Sigmoid function and Fourier function. Thus, the output vector of the SLFN with respect to x_j, $\vec{o}(x_j)$ is:

$$\vec{o}(x_j) = \sum_{k=1}^{L} \beta_k h_k(x_j) \tag{3}$$

where L is the number of hidden nodes and β_k is the output weight of node k in the hidden layer to the output layer. Eq. 3 can be re-written as:

$$\mathbf{H}\beta = \mathbf{O},$$

where $\mathbf{O}$ is the output matrix of the SLFN, β is the weight matrix of the hidden layer nodes to the output layer nodes and $\mathbf{H}$ is the the hidden layer output matrix. $\mathbf{H}$, given N training samples is composed of the following:

$$\mathbf{H} = \begin{bmatrix} \vec{h}(x_1) \\ \vdots \\ \vec{h}(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \ldots & h_L(x_1) \\ \vdots & \vdots & \vdots \\ h_1(x_N) & \ldots & h_L(x_N) \end{bmatrix} = \begin{bmatrix} G(w_1, b_1, x_1) & \ldots & G(w_L, b_L, x_1) \\ \vdots & \vdots & \vdots \\ G(w_1, b_1, x_N) & \ldots & G(w_L, b_L, x_N) \end{bmatrix}.$$

The purpose of the SLFN is to minimize the cost function $||\mathbf{O} - \mathbf{T}||$ where $\mathbf{T}$ is the target label matrix for the respective samples. This allows us to approximate the target class as accurately as possible, given the samples. However, conventional methods for building and training neural networks involves gradient-based learning algorithms and the tuning of parameters e.g. the learning rate and the number of iterations, which are time-consuming. ELM, on the other hand, is claimed to be able to learn at a much faster speed. It starts by randomly assigning values to the weight and bias parameters of the input nodes to the hidden layer nodes, w_k and b_k. Then, the hidden layer output matrix, $\mathbf{H}$ of the SLFN is calculated and finally, the output weight of the hidden layer nodes to the output layer, β can be mathematically determined by finding the least-squares solutions of the linear system:

$$\beta = \mathbf{H}^{\dagger}\mathbf{T}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of $\mathbf{H}$. This removes the need for the tuning of parameters and slow learning algorithms, thus speeding up the training process of the SLFN.

Wang et al. [81] used this ELM method to evaluate the uncertainty in sensor measurements. The ELM model the process in which the input values not only consist of raw sensor data but also the system state, which affects the “ground truth” value. The paper states that the approximation of “ground truth” value can be calculated as $p(\hat{x}_t|s_t)p(s_t|x_t)$

where $p(\hat{x}_t|s_t)$ denotes the occurrence probability of an individual measurement conditioned on another, x_t and $\hat{x}_t$ are the observed measurement and approximate "ground truth" respectively and s_t is the system process state. Thus, using two networks of ELM, a measurement model that represents the measurements and system state is built in the first ELM to find $p(s_t|x_t)$. Then, the second ELM is established to estimate $p(\hat{x}_t|s_t)$ given the estimated part quality from the first ELM.

Ensemble classifiers Rahman et al. [79] proposed a supervised classification framework for automatic quality assessment through ensemble Decision Trees and Bayesian Network classifiers. The uncertainty in the data is represented as quality flags, e.g. "Good data", "Bad data", "Probably good data" and "Bad but correctable data". The classifier is trained on training data labelled with quality flags by domain experts. However, since class imbalance exists (a small number of anomalies), it is trained on under-sampled data which is sampled on clusters obtained from k-means clustering. The sampling from the clusters formed by the k-means clustering algorithm ensures that it is representative of the significant areas of the data. The decisions by the base classifiers are fused using a majority voting fusion rule based on the mode of the decisions.

Ontology/knowledge-based systems Kuka and Nicklas [26] proposed a framework for quality indicators for inaccurate and incomplete data, and also other quality indicators such as inconsistent data and timeliness using ontology (Sensor Network Ontology) to enrich sensor data streams by propagating quality semantics. The paper defines the quality indicators as:

1. Timeliness—the timestamp (start timestamp of the measured data to the time when the data reaches the system) divided by the frequency of sensing device.
2. Accuracy—the variance of the observation and uncertainty, modelled as a mixture of Gaussian models with mean and variance.
3. Completeness—the number of attribute values that are not null, for probabilistic attributes i.e. ones with the Accuracy property, Cumulative distribution functions are used.
4. Consistency—the similarity of two observations measuring the same variable from different sensing devices are valid at the same time.

Bamgboye et al. [25] also suggested a software architecture solution based on semantic technology for Smart Spaces applications, a part of the Smart Cities ecosystem, which improves data stream quality by quantifying inaccurate and incomplete data, (along with other DQ dimensions e.g. inconsistent data, availability, and timeliness) based on expert knowledge. The semantic framework aims at homogenizing, annotating and reasoning over the sensor data and it consists of 4 layers:

1. Data abstraction layer—collects raw data from sensor devices using the Global Sensor Network middleware and uses static knowledge base to perform filtering of data points with quality related problems.

Table 4 Methods for error correction (RQ3), along with its addressed errors, respective papers, and the total number of papers that proposed that method

Method	Errors addressed	Papers	Total
Association rule mining	Missing data	[61, 62, 64, 66]	4
Clustering	Missing data	[65]	1
k-Nearest Neighbour	Missing data	[9]	1
Singular value decomposition	Missing data	[67]	1
Empirical mode decomposition	Noise	[76]	1
Savitzky–Golay filter and multivariate thresholding	Noise	[77]	1
Hybrid methods			
Clustering and probabilistic matrix factorization	Missing data	[63]	1

2. Modelling and integration layer—Provides a platform for interoperability and integration for the heterogeneous data from different types of sensor devices by implementing domain ontology from semantic sensor network.
3. Reasoning layer—consists of predefined rules obtained from domain expert knowledge and semantically annotated data streams from the second layer to perform reasoning.
4. Application layer—contains application programs that rely on the sensor generated data streams and relies on the previous lower layers.

Methods for correcting errors in sensor data

Out of the 57 publications found in this systematic review, there are ten publications which presented approaches focused on correcting errors in sensor data. The methods suggested focus only on correcting errors such as missing data and noise but do not attempt to detect or quantify them. The correctional methods can be termed as *missing data imputation*, which tries to estimate sensor measurement values that are missing and *de-noising*, which tries to remove the noise associated with a measurement signal. Table 4 shows the different existing methods proposed to correct sensor data errors (RQ3), which consists of missing data and noise, along with the errors addressed, the corresponding papers that proposed the method and the total number of papers. The most common method for missing data imputation is Association Rule Mining, which is addressed by half of the papers that deals with missing data estimation techniques. There are also two clustering techniques presented, though one of them is a hybrid approach with Probabilistic Matrix Factorization. There are also only two de-noising methods found in the selected studies, which is the Empirical Mode Decomposition and Savitzky–Golay Filter.

Missing data imputation

For missing data error, Association rule mining, Clustering, k-Nearest Neighbour, and singular value decomposition solutions have been proposed to estimate the missing sensor values.

Association rule mining Gruenwald et al. [64] came up with an association rule mining approach called FARM (Freshness Association Rule Mining) to estimate missing values in sensor data. The central idea of this approach is that more recent sensor data values should have a higher contribution to the association rule, that will be used for imputing missing data at a specific point in time. This is because usually, the current state of a sensed physical environment is more dependent on its nearest previous states, rather than historical states that are obtained long ago. For this purpose, round weights are added to each row of data. The FARM approach also uses the Apriori Association Rule mining algorithm to estimate the missing sensor value based on the weighted average of the current reading of the sensors related to the sensor with the missing readings (obtained by the Association Rule Mining). Since the freshness concept is introduced, the weighted support and confidence measure is modified to the following:

$$sup_w(A \implies B) = \frac{\sum round\ weights\ where\ A\ and\ B\ report\ the\ same\ state, e}{\sum round\ weights},$$
$$conf_w(A \implies B) = \frac{\sum round\ weights\ where\ A\ and\ B\ report\ the\ same\ state, e}{\sum round\ weights\ where\ e\ is\ reported\ by\ X}.$$

In 2009, Chok and Gruenwald [61] refined the approach to cope with the complexity of data streaming environments using a MASTER-Tree data structure. Moreover, Wang et al. [66] proposed the Time-Space relationship and Association Rule Mining method for interpolating missing data in activity recognition applications. This differs from the FARM approach [64] as it incorporates the spatial correlation between sensor nodes using Pearson's correlation coefficient. This reduces complexity for the Association Rule Mining algorithm as it only needs to search for rules from sensors that have a correlation coefficient above a certain user-defined threshold.

D'Aniello et al. [62] incorporated association rule mining in their virtual sensor framework to impute missing data values. Their framework also uses ontology to represent sensors and data quality along with fuzzy logic to evaluate the quality of data received. For example, the sensor is characterized by several quality criteria such as those declared in the manufacturer specifications e.g. accuracy, precision and time since last calibration. Users can also specify their quality requirements, e.g. requiring response time $\leq$ 30 ms $\pm$ 2 ms, which can be expressed in fuzzy sets, e.g. *low* response time. The virtual sensor thus attempts to meet those quality requirements of the users by providing the real reading if it meets those criteria or the reconstructed value if the value is missing or if it does not meet the criteria. The reconstructed value is computed using association rule mining, which exploits the spatio-temporal correlation among sensor readings to estimate missing data.

Clustering Tang et al. [65] introduced a method for missing data imputation using fuzzy C-means clustering, which has its parameter optimized using Genetic Algorithm. The fuzzy C-means clustering algorithm aims to classify data into different clusters to maximize their similarity. The weekly traffic volume data from sensors are analysed and converted from a vector-based data structure into a matrix data structure. The Fuzzy C-means clustering model is then built using Genetic Algorithm to optimize the membership degrees and cluster centers.

k-Nearest Neighbour Li and Parker [9] proposed an imputation technique for missing data using the Nearest Neighbour approach which takes advantage of spatio-temporal correlations in the sensor data. The method uses a *k*d-tree structure to search for the nearest neighbours, formed using weighted Euclidean metric which takes into account the percentage of missing data for each sensor. Then, the algorithm searches the tree to find the nearest neighbours and impute missing values based on the values obtained from its neighbours (hot deck imputation).

Singular value decomposition Xu et al. [67] presented a mathematical approach for recovering missing data by representing the spatio-temporal sensor data as a multi-dimensional tensor (tensors are a multi-dimensional extension of a matrix) and introduced a tensor-based recovery method i.e. tensor singular value decomposition (t-SVD) to recover the missing values. One of the advantages of this method is that it does not require non-missing training data. The spatial correlations between the sensors are firstly obtained using Nearest Neighbour search, which forms a two-dimensional, $lat \times long$ matrix, which represents its latitude and longitude. Apart from the spatial correlation, the temporal correlation is also represented in the same tensor. This is done by either formulating it as a three-order, $lat \times long \times hour$ tensor, or a four-order, $lat \times long \times hour \times day$ tensor which includes the models of the same hours in a day, or a five-order, $lat \times long \times hour \times day \times week$ tensor, which models the similarity of the same hours in different days, and the same day in different weeks. After having the appropriate tensor representation of the data, t-SVD is applied, which recovers the missing values.

Hybrid methods Fekade et al. [63] proposed a *k-means clustering* and *Probabilistic Matrix Factorization* (PMF) approach to recover missing values. Firstly, k-means clustering is done to divide the data into clusters, and within each cluster, PMF is applied. PMF decomposes a single matrix into a product of two matrices, which has the property to obtain the original matrix by computing the product of two matrices, thus enables the recovery of missing values in the original matrix.

De-noising

Other than handling missing data, there are two publications found in the 57 selected studies that presented noise error correction (de-noising).

Signal processing Omitaomu et al. [76] suggested an approach for de-noising sensor signals using the shrinkage method (thresholding) to de-noise high-frequency intrinsic mode functions (IMF). IMF is an oscillatory signal which is a subset of the frequency components from the original signal. It can be obtained by applying Empirical Mode Decomposition, which forms low-frequency and high-frequency IMFs. They can then be separated by mutual information. The method studied in this paper only considers applications with signals that are corrupted by high-frequency noise, whereby de-noising the low-frequency IMF can lead to loss of signal information.

Table 5 Methods combining error detection and correction (RQ2 and RQ3), along with its addressed errors, respective papers, and the total number of papers that proposed that method

Method	Errors addressed	Papers	Total
Principal component analysis	Outliers, bias, drift, constant values, noise, stuck-at-zero	[46, 55]	2
Artificial neural network	Outliers, bias	[41, 43]	2
Bayesian network	Outliers, missing data	[7, 38]	2
Grey prediction model	Outliers, bias, constant values, stuck-at-zero	[30]	1
Dempster–Shafer theory	Uncertainty	[80]	1
Calibration-based method	Bias, drift, noise, stuck-at-zero	[73]	1
Hybrid methods			
Principal component analysis-based methods	Outliers, bias, drift, noise, constant values, stuck-at-zero	[60, 72, 74]	3
Kalman filter-based methods	Outliers, bias, drift, missing data	[31, 51]	2
Dempster–Shafer theory & Ontology	Uncertainty (inaccurate data), missing data (incomplete data)	[68]	1

Savitzky–Golay filter and multivariate thresholding Sadıkoglu and Kavalcıoğlu [77] presented a de-noising approach for a healthcare application, specifically continuous glucose monitoring systems, using Savitzky–Golay Filter and Simple Multivariate Thresholding. The Savitzky–Golay Filter is a method of data smoothing based on local least-squares polynomial approximation whereas the Simple Multivariate Thresholding is a multivariate extension of a wavelet de-noising strategy which combines univariate (one-dimensional) wavelets de-noising algorithms and PCA for dimensionality reduction.

Methods for detecting and correcting errors in sensor data

There are 15 out of 57 publications that answer both RQ2 and RQ3 simultaneously by detecting the error and correcting them. They are usually termed as *fault detection, isolation, and recovery*. Those introduced methods usually come in the form of building a normal behaviour model of the system and comparing the new observed values with the normal model. If the current observed data is significantly different from the estimated value, it is identified as anomalous and is imputed with the estimated value from the model. Table 5 shows the different existing methods presented to detect and correct sensor data errors (RQ2 and RQ3), along with the errors addressed, the respective papers that proposed the method and the total number of papers. The six hybrid methods suggested for fault detection and correction can be classified into PCA-based hybrid methods, Kalman filter-based hybrid methods, and Dempster–Shafer Theory-based methods. It is seen that PCA-based methods are most commonly found in this area, which consists of one-third of the total papers addressing the fault detection, isolation, and recovery problem.

Fault detection, isolation and recovery

The following are the different approaches proposed to detect, isolate (identify) and correct errors in sensor data.

Principal component analysis Liu et al. [55] presented a PCA-based self-validating sensor approach for wastewater treatment plants which is able to identify faulty sensors before soft sensor prediction, using the Squared Prediction Error (Q-statistic) and Sensor Validity Index (SVI) [27]. The reconstructed vector, $\vec{x}^*$ of a faulty sensor data can be obtained by subtracting the fault from the observed data, $\vec{x}$:

$$\vec{x}_i^* = \vec{x} - f_i \vec{\epsilon}_i \,, \tag{4}$$

where f_i is the magnitude of the fault and $\vec{\epsilon}_i$ is the direction of the fault. Thus, the goal is to find f_i such that Eq. 4 is most consistent with the PCA model. The approach is further refined in [46] where another variation of PCA called the Variable Bayesian PCA is suggested to handle missing data in the training set, which can cause over-fitting and locally bad optimal solutions.

Artificial neural network Huang [43] introduced a technique for sensor fault e.g. outliers and bias diagnosis and reconstruction using auto-associative neural networks (AANN) which learn the internal relationship between all inputs by encoding (compressing) and decoding (decompressing) the data. Moreover, Hou et al. [41] came up with a technique for sensor fault diagnosis and validation using rough sets for pre-processing (for dimensionality reduction and to learn classification rules) and artificial neural networks to learn the normal behaviour model.

Bayesian network Dereszynski and Dietterich [38] proposed a data imputation method for missing values and anomalies based on a dynamic Bayesian network which learns the normal behaviour model of sensor measurements. The discrepancy between the current estimate from the Bayesian network model and the current observed reading detects if the reading is anomalous. Since a normal static Bayesian network only models the spatial correlation in the dataset, a dynamic Bayesian network is used to also incorporate the temporal correlations, since environmental data tend to be temporally correlated e.g. patterns for the 24-h cycle is relatively similar. It relates variables to each other over adjacent time steps. Zhang et al. [7], also suggested a data reconstruction technique for missing data and inaccurate values in medical body sensor networks by learning a Bayesian network. The Bayesian network learns the probabilistic graphical model and estimates the sensor value by calculating its conditional probability.

Grey prediction model Chen et al. [30] proposed a self-validating strategy for multifunctional sensors using the Grey Bootstrap model (GM(1,1) with bootstrap) which produces a prediction model. Current observations will then be compared to the predicted value to detect, isolate, and recover faults such as outliers, bias, constant value, and near-zero values. Bootstrapping can be done by drawing random samples from the dataset with replacement to generate B bootstrap samples and calculate the estimate for each resample, which gives the approximation of uncertainty. The bootstrap method allows the uncertainty to be estimated without having prior information about the probability distribution of the measurements. For each bootstrap sample, a grey predictive model, GM(1,1) is used to predict the next value.

Dempster–Shafer theory Richter [80] presented a method for assessing uncertainty and increasing reliability for context-based applications (activity recognition). The reliability assessment is done via mean squared error and to increase reliability, data fusion by Dempster–Shafer theory of evidence is used in which measurement result is combined with other sensor events that have higher reliability and are spatio-temporally related. The Dempster–Shafer theory combines evidence gathered from multiple sources to derive a new degree of belief, also known as belief mass, by data fusion and calculates the confidence interval which includes the exact probability without needing prior information. Thus, it provides more flexibility compared to Bayesian Networks as it requires weaker conditions.

Calibration-based method Yu and Li [73] introduced an online in-situ calibration technique based on a calibration method. The calibration technique corrects faults such as bias, drifts, noise, and sensor failure. An environment evaluation is firstly carried out, in which a benchmark is established and measured values can be compared to the benchmark and calibrated via a mapping to the benchmark. However, this method requires an accurate environment evaluation.

Principal component analysis-based hybrid methods Wang et al. [74] presented methods for online blind detection and automatic calibration of sensor drifts via signal space projection using *Principal component analysis* to learn the normal sensor behaviour model and detect the sensor drifts. Then, *Kalman filter* is applied to estimate the sensor drift value and the drift value is subtracted from the sensor reading to give a better estimate of the true value. Yang et al. [60] suggested a data validation technique to detect, identify and correct faults such as bias, drifts, and impacts in a multifunctional sensor. The technique separates (using Maximal Information Coefficient) the variables into independent and dependent variables. Different fault detection, identification, and correction techniques are used for the two types of variables. For independent variables, *k-Nearest Neighbour* is used for fault detection and identification and a *Grey Predictive Model* GM(1,1) is used for fault correction. For related variables, kernel *Principal component analysis* is used for fault detection. *Iterative Reconstruction-Based Contribution* is used for fault identification which assumes that the sensors are faulty and iteratively reconstruct the data until its Squared Prediction Error (Q-statistics) is below a certain threshold and finally, variables in the estimated fault direction are deemed to be faulty. For fault correction of the related variables, *fuzzy similarity* is used, which involves reconstructing the faulty variable based on the relationships between the related variables. Furthermore, Uren et al. [72] proposed a PCA-based sensor fault detection, isolation, and reconstruction for bias, drifts, noise, constant value, and stuck-at-zero errors. For fault detection, non-temporal parity space is used to check for inconsistencies among a set of redundant sensors. With the assumption that not all sensors may fail simultaneously in a particular channel, the non-temporal parity space technique compares and validates the sensor measurements with a set of redundant measurements. An estimate is obtained from the most consistent subset of redundant measurements of a process variable and the faulty sensor can be identified via parity checks. *Fuzzy rule* base is then used for fault isolation and *Principal Component Analysis* is used to model and reconstruct the sensor measurements.

Table 6 Domains of sensor data quality application from the 57 selected papers, along with its respective papers and total number of papers that solve sensor data errors in that domain

Domain	Papers	Total
General e.g. WSNs, IoT, streaming data	[26, 27, 33–37, 39, 46, 48–53, 57, 58, 60–64, 67–69, 74–76, 78]	29
Industrial processes e.g. Chemical gas process monitoring, power plants, part injection molding	[30, 43, 44, 70–72, 81]	7
Environmental sensing e.g. air quality monitoring, marine environment, soil moisture	[32, 38, 40, 47, 56, 79]	6
Smart city e.g. Smart Spaces, Smart Grid, Wastewater treatment, Traffic flow	[25, 45, 46, 54, 55, 65]	6
Healthcare e.g. body sensor networks, artificial pancreas, continuous glucose monitor	[7, 31, 59, 77]	4
HVAC systems	[41, 42, 73]	3
Context-based application / activity recognition	[66, 80]	2

Kalman filter-based hybrid methods Solomakhina et al. [51] suggested an approach for detecting and correcting anomalous sensor data using *Kalman Filter*, Autoregressive Integrated Moving Average (*ARIMA*), smoothing operators and *knowledge-based systems*. The errors, e.g. missing data and outliers, are detected using the ARIMA and Kalman filter, and the Knowledge-based system is consulted to confirm the error. In the healthcare domain, Feng et al. [31] also introduced a Kalman filter-based hybrid approach for sensor fault identification and correction for missing data, bias, drifts and outliers. The method uses *Outlier Robust Kalman filter* (ORKF) and *locally-weighted partial least squares* (LW-PLS) for artificial pancreas control systems. Both algorithms are used to model the normal sensor behaviour to provide a more robust fault detection since one might report an error, but the other might not. The ORKF has the advantage of fast detection and online auto-smoothing i.e. de-noising ability, which works well for short-duration errors such as outliers. For long-duration errors, such as drifts, LW-PLS is more advantageous as it is based on historical data. The errors are then replaced by the estimated value which has the highest performance score e.g. model accuracy, smoothness from the models.

Dempster–Shafer theory-based hybrid method To evaluate the quality of sensor data based on inaccurate and incomplete data, (as well as inconsistent data and timeliness), Hermans et al. [68] presented a framework using *heuristics* in the local and cluster heads, and an inference engine in the cluster head which fuses correlated sensor readings using *Dempster–Shafer theory* to arrive at a more accurate estimate.

Types of domains

The domains of the applications (RQ4) in the 57 selected papers are extracted and the results are tabulated in Table 6. About half of the publications suggested methods that generally apply to WSNs or IoT applications without a specific application domain. However, some publications solve data quality problems in specific areas such as industrial processes, environmental sensing, and smart city solutions. Other domains also

Table 7 Real-world publicly available datasets and its respective domains, along with the respective papers and total number of papers which used the datasets for performance evaluation

Dataset	Domain	Papers	Total
SensorScope (GSB, LUCE, FishNet)	Environmental sensing	[35, 57, 58] [38] (GSB and FishNet) [48] (GSB and LUCE)	7
Intel Berkeley	Environmental sensing	[35, 36, 39, 48, 62, 63]	6
UCI machine learning repository water treatment plant dataset	Smart city (wastewater treatment)	[46, 55]	2
Numenta anomaly benchmark	General (streaming data)	[34]	1
Networked aquatic microbial observing system (NAMOS)	Environmental sensing (marine environment)	[48]	1
TasMAN Sullivans Cove Marine	Environmental sensing (marine environment)	[79]	1
MERLSense	Environmental sensing	[66]	1
Caltrans PeMS traffic monitoring	Smart city (traffic flow monitoring)	[9]	1
PhysioNet	Healthcare	[7]	1

include healthcare, heating, ventilation, and air conditioning (HVAC) systems, and activity recognition applications.

There are several publicly available datasets that are used in method evaluation and are domain-specific. The nine publicly available datasets are: SensorScope dataset [93, 94], which includes the Grand St. Bernard (GSB), FishNet and Lausanne Urban Canopy Experiment (LUCE) deployments, Intel Berkeley dataset [95], University of California Irvine (UCI) Machine Learning Repository's water treatment plant dataset [96], Networked aquatic microbial observing system (NAMOS) dataset [97] from the University of Southern California, numenta anomaly benchmark dataset (NAB) [88, 98], California's Department of Transportation (Caltrans) Performance Measurement System (PeMS) traffic monitoring dataset [99], Tasmania Marine Analysis Network (TasMAN) Sullivans Cove CSIRO Wharf marine dataset [100], US Mitsubishi Electric Research Laboratories MERLSense dataset [101, 102], and PhysioNet [103]. The datasets and papers that used them for method evaluation and the total number of papers are listed in Table 7. These datasets are last searched on May 8th 2019.

The following are brief descriptions of the publicly available datasets. SensorScope has deployed many outdoor networks for environmental monitoring which produced datasets, three of which have been used by seven of the selected papers for method evaluation. Those publications include ones without a specific domain (general papers) such as [57] or papers that are environmental sensing domain-specific such as [38]. The GSB SensorScope network has been deployed at the Grand St. Bernard pass, located at the border of Switzerland and Italy, in late 2007 for approximately 1 month (September–October). It comprises of 23 stations. Another SensorScope network is the FishNet deployment, which is deployed 1 month before GSB (August–September) for around a month as well, but only with six stations. It is used to monitor a river to improve its quality. The last SensorScope dataset seen in the selected literature is the LUCE dataset, which is a more extensive dataset, with 97 stations deployed for almost a year from July 2006 to May 2007, which aims to understand the atmospheric behaviour in urban environments better.

Another dataset related to indoor environmental monitoring is the Intel Berkeley Research Lab dataset. There are six papers in total that have used this dataset to test their proposed methods. The Intel Lab data has 54 sensors deployed indoors in the lab itself, collecting data about the environment such as the temperature, humidity, and light. It also consists of the date, time, epoch, sensor ID, and voltage values. The units for the data types are also specified, where the temperature is recorded in degrees Celsius (^oC), the humidity is in percentage (%), which is the relative humidity, and the light intensity is measured in Lux. The dataset consists of 2.3 million readings, obtained every 30 seconds from the sensors for about a month (February 28th to April 5th 2004). MERLSense is also another environmental sensing dataset, where it captured and recorded motion data. It is also deployed in a research lab, and data of the people working in the lab is collected for 2 years from March 2006 and March 2008, totalling up to 50 million raw records from 200 sensors. However, Wang et al. [66] has used the temperature attribute to evaluate their missing data imputation technique.

The NAMOS and TasMAN datasets are both marine datasets, collected to monitor marine environments. NAMOS consists of several datasets from devices deployed by the University of Southern California at different locations e.g. buoys, boats and weather stations around California. It is one of the three datasets, along with the Intel Lab and SensorScope LUCE datasets that Rassam et al. [48] used to evaluate their method for outlier detection. The NAMOS dataset used is from the buoy no. 103 collected in August 2006 in Lake Fulmor. The TasMAN dataset, on the other hand, is from Sullivans Cove, Hobart, Tasmania. The dataset consists of the seawater temperature and conductivity from February 2008 to July 2012. Numenta anomaly benchmark (NAB) is an open-sourced benchmark for evaluating techniques for anomaly detection for streaming data. It provides numerous real-world streaming datasets such as Amazon's AWS server metrics, online advertisement clicking rates, temperature sensing, and traffic monitoring datasets. The datasets provided by NAB are the only ones, among the other datasets found in this systematic review, complete with labelled streaming data comprising of normal and erroneous measurements e.g. spatio-temporal outliers, noise and drift. It also has a novel scoring system called the NAB score, which takes into account true positives, true negatives, false positives, false negatives, and windows to reward early detection.

Moreover, another publicly available dataset comes from the University of California, Irvine (UCI) Machine Learning Repository. The repository contains a wide range of 468 different datasets, from healthcare to games, robotics to social science, environmental sensing and many more. There are two of the selected papers [46, 55] that have used water treatment plant dataset from this repository. It has 527 samples and 38 attributes, which consists of the daily measurement from sensors in an urban wastewater treatment plant. It is a relatively complex system, where the aim is to predict faults through the operational state of the process. Another publicly available smart city dataset comes from California's Department of Transportation, who released their traffic monitoring datasets. However, a user has to sign up for a free

Table 8 Types of datasets used in method evaluation and their availability online or reproducibility, the total number of datasets used for each type of dataset and the respective papers and the total number of papers that used those datasets

Dataset type	Availability	No. datasets	Papers	No. papers
Real-world datasets	Published and currently available	21	[7, 9, 34–36, 38, 39, 46, 48, 55, 57, 58, 62, 63, 66, 79]	16
	Unpublished or currently not available	33	[9, 30, 31, 33, 35–37, 40–42, 44, 45, 47, 49, 50, 52–54, 56, 60, 61, 64, 65, 67–70, 72, 73, 75, 76, 81]	32
Simulated datasets	Published or reproducible	2	[46, 54]	2
	Not reproducible	16	[35, 37, 39, 43, 47, 51, 57–59, 69, 71, 74, 76–78]	15

account to access those datasets. It provides an extensive traffic monitoring dataset, where users can choose to obtain data from up to almost 100 freeways in California gathered by 18,305 stations. The website also provides real-time information displayed on a dashboard. The last publicly available dataset found to be used for method evaluation for one of the 57 selected papers is PhysioNet. It provides healthcare-based sets of data, which is split into two categories: clinical databases and waveform databases. The former provides data such as demographics, images and vital sign measurements where Zhanget al. [7] used one of the datasets, whereas the latter presents a digitalized signal or waveforms of physiologic data, such as the heart monitoring electrocardiogram device signal.

Discussion

In this section, we discuss the challenges found through the systematic review, which affects the comparability of methods introduced in this research area. The evaluation of the performances of methods presented in the selected studies are done on a wide range of datasets and have different dataset pre-processing conditions. This makes it impossible to compare the efficiency of these methods just by reading the respective publications. Furthermore, there are various evaluation metrics used in the literature and even within the same problem, e.g. outlier detection, the evaluation metrics used are different. This shows that there is no generally accepted way of comparing different methods. An analysis of the problems is detailed in the subsections that follow. "Datasets and error imputation and labelling" section discusses the different datasets used and their availability online or reproducibility, and the preparation or pre-processing of the dataset which includes error introduction for evaluating the methods. "Evaluation metrics" section, on the other hand, details the different evaluation metrics used and the situations they are used in.

Datasets and error imputation and labelling

There are many different datasets used in the performance evaluation of methods found in the literature. They can be categorized into two types of datasets: real-world datasets and simulated datasets. Real-world datasets consist of data from real-world experiments or deployments, whereas simulated datasets contain data that have been synthetically produced. Within those two categories, the datasets can be further split into published or unpublished datasets. Published datasets are datasets that are currently publicly available (last checked on May 8th 2019) or reproducible datasets, by having the dataset itself or source code published online. The published datasets are described in "Types of domains" section with respect to their domains. Unpublished datasets refer to the datasets that are not currently available publicly or cannot be reproduced. Table 8 shows the different types of datasets and the number of papers that used them for evaluation. Out of the 57 final selected publications, 52 publications have proper validation and from these, there are a total of 72 datasets used for evaluating the introduced algorithms. From these 72 datasets, there are 54 real-world datasets of which only 21 of them are published datasets. Furthermore, among the 18 simulated datasets, only two can be reproduced as the simulator is publicly available. Note, that some papers evaluated their methods on more than one dataset, such as the work of Bosman et al. [35], who used four datasets: the published real-world datasets Intel Lab and SensorScope GSB, an unpublished real-world dataset, and a simulated dataset.

From the 72 datasets used for method evaluation, it is seen that around 68% of the datasets are not published nor reproducible, consisting of both real-world and simulated datasets. This makes it hard for the comparison of different methods in this research area. Besides that, even for literature working on publicly available datasets, they have different techniques of imputing errors to evaluate their suggested methods. For example, in anomaly detection, Bosman et al. [35] labelled the errors in the Intel Lab and SensorScope GSB datasets using a semi-automated approach. Anomalies were identified by heuristics (e.g. a value that is not changing for over ten samples is labelled as a constant-value error), which will then be corrected manually by a person. Rassam et al. [48], on the other hand, also used the Intel Lab dataset and the SensorScope dataset (LUCE, NAMOS) to evaluate their proposed method but have a different heuristics of histogram-based labelling. They assessed their solution on simulated errors whereby they artificially injected 100 anomalies. For missing data imputation, the Intel Lab dataset has also been used by D'Aniello et al. [62] and Fekade et al. [63], where the former simulated the missing errors at 5%, 10%, 20%, 30%, 40% and 50% rates and the latter simulated the missing error by making 10% of the total data empty.

Other than having different error injection and labelling methods, though even if two publications might use the same online dataset, it is still not directly comparable as they might have pre-processed the dataset. For example, Bosman et al. [36] and Fawzy et al. [39] both used the entire Intel lab dataset for evaluation, though with different error introduction techniques, whereas Rassam et al. [48] only used 3 out of the total 54 sensor nodes. D'Aniello et al. [62] also removed known errors from the dataset before proceeding to test their missing data imputation methods on a subset (March 1st–14th) of the Intel Lab dataset. Although the NAB is a unified benchmark that provides publicly available datasets complete with labelled errors, it is only for the comparison of anomaly detection algorithms. It does not provide a benchmarking system for missing data imputation and fault correction, which is the other two main types of errors found in the literature. There is also only one publication [34] among the 52 publications with validation that has used this benchmarking system.

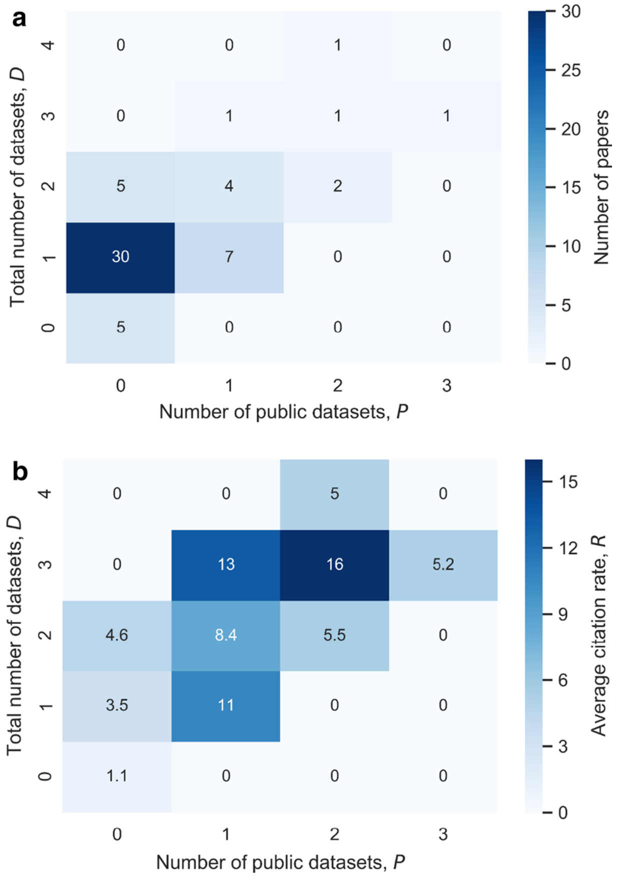

Fig. 5 Heatmap of dataset availability. The two heatmaps describing the problem of availability of the datasets used in sensor data quality literature. Both heatmaps show the number of available datasets, *P* against the total number of datasets, *D* used for method evaluation in the 57 selected papers where the number in each cell represents: **a** the number of papers that have used *P* publicly available datasets out of its *D* total number of datasets for method evaluation and **b** the average citation rate for the papers that have used *P* publicly available datasets out of its *D* total number of datasets for method evaluation

In order to analyse the problem of the availability of the datasets used in the literature, we are introducing two data set metrics, which are retrieved for each of the 57 reviewed publications. The first metric is the number P_k of publicly available data sets, which have been used for evaluating purposes in the kth paper. The second metric is the total number of data sets D_k, which have been used for the evaluation of algorithms in the kth paper. Consequently, the difference $(D_k - P_k)$ is the number of datasets, which have been evaluated in the kth paper but are not publicly available. However, up to this point, our model does not take into account the possible influence of open access publications of the respective papers on its citation rate.

Figure 5a shows a heatmap of the number of publicly available datasets P_k against the total number of datasets D_k used in the sensor data quality literature. The heatmap visualizes the joint distribution of $P = (P_1, \ldots, P_{57})$ and $D = (D_1, \ldots, D_{57})$: The numbers in each cell corresponds to the number of publications that have used the respective number of publicly available datasets P and the total number of datasets D to evaluate their methods. The darker colour (higher numbers) in the lower quadrant shows that the majority of the reviewed sensor data quality publications evaluate their methods on fewer datasets and more importantly, on datasets that are not publicly available. However, a problem found through this systematic review, which is the direct comparability of the methods introduced, can only be solved if researchers in the research area evaluate their techniques on the same datasets (given the same error injection and pre-processing conditions). Thus, having used publicly available datasets might prove to be beneficial to the research area. This prompts for a need to further analyse the effects of using publicly available datasets for method evaluation.

In order to do so, we are analysing the citation rate R_k of the reviewed publications. The citation rate R_k is computed as the quotient of the number of citations of the kth paper and its years since publication. The number of citations for each publication has been obtained from Google Scholar[5] on April 5th 2019. Google scholar was chosen because it can be accessed without a license fee. Note, that Google Scholar's citation count includes citations from various sources including self-citations and preprint repositories. Thus, the citation count might differ from other citation databases, e.g. Web of Science. The citation rates R_k are binned with respect to their total number of datasets D_i and number of publicly available data sets P_j and are averaged for each bin:

$$\overline{R}_{i,j} = \frac{\sum_{k=1}^{57} \mathbb{1}_{D_k=i}\mathbb{1}_{P_k=j} R_k}{\sum_{k=1}^{57} \mathbb{1}_{D_k=i}\mathbb{1}_{P_k=j}} \text{ with } \mathbb{1}_{a=b} = \begin{cases} 1: & a = b, \\ 0: & a \neq b. \end{cases} \tag{5}$$

From this observation, the heatmap in Fig. 5b is plotted to study the effects of using publicly or non-publicly available datasets on the citation rate of a publication. The citation rate is used to study the effects of using publicly or non-publicly available datasets as a high citation rate might imply that researchers working on the same research area can compare their results with those studies using the same dataset. If the dataset is not publicly available, it makes it hard for comparison. The heatmap shows that the average citation rate for the literature involving publicly available datasets tends to be higher. To confirm this observation, a Bayesian analysis [104] is carried out to test if there is a significant difference in the citation rate between two groups: the *available group*, which consists of papers that evaluated their methods on publicly available datasets and the *non-available group*, which consists of papers that evaluated their methods on non-publicly available datasets.

Shown in Fig. 6, the Bayesian estimation is carried out using the Python module `PyMC` [105], which is designed to implement Bayesian statistical models. A Bayesian estimation is done instead of the classical t-test, as it shows the complete distributional information, i.e. the probability of every possible difference of means and every possible difference of standard deviations which allows the *estimation* of the difference between the two groups rather than simply *testing* whether the two groups are different based on the observed data [104]. Figure 6a, b show the posterior distribution of the mean citation rates for both groups, i.e. the available group and the non-available group. The

[5] https://scholar.google.com.

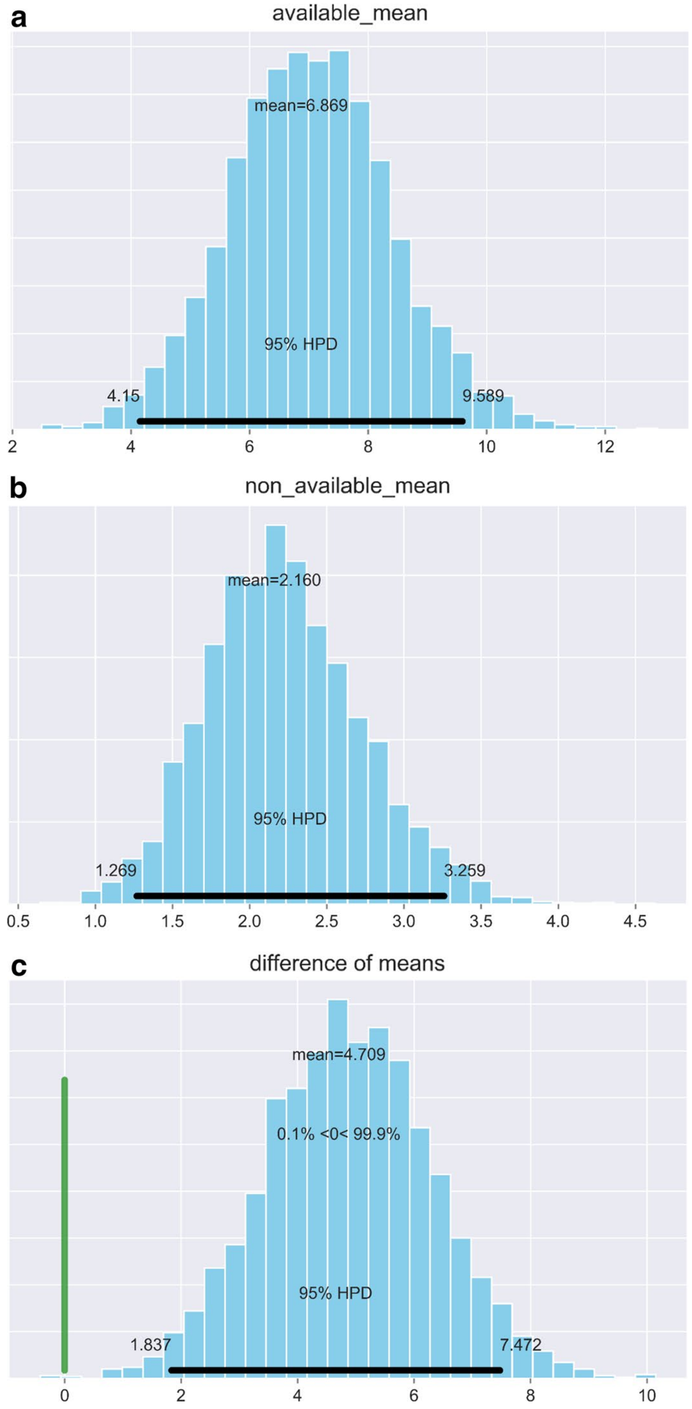

Fig. 6 Bayesian analysis. Bayesian analysis which shows a significant difference in the citation rate between the *available group*, and the *non-available group*: **a** the mean citation rate, 6.79 of the *available* group which consists of the collection of papers that used publicly available datasets to evaluate their methods whereas **b** the mean citation rate, 2.16 of the *non-available* group which involves the group of papers that did not use publicly available datasets for method evaluation. **c** The difference of means from both groups. The papers in the *available* group has a 99.9% posterior probability of having higher number of citations compared to the papers from the *non-available* group

Table 9 Confusion matrix where the positive class are faults and the negative class are normal data points

	Actual positive	Actual negative
Predicted positive	True positive (TP)	False positive (FP)
Predicted negative	False negative (FN)	True negative (TN)

Thus, *TP* faults correctly predicted as faults, *FP* normal data point incorrectly predicted as fault (Type I error), *FN* fault incorrectly predicted as normal data point (Type II error) and *TN* normal data point correctly predicted as normal data point

Table 10 Types of performance measures used in method evaluation for the 39 papers which has quantitative performance values and their respective formulas, papers and total number of papers, where TP = true positive, TN = true negative, FP = false positive, FN = false negative, x_i = observed value or ground truth of sample i, $\hat{x}_i$ = predicted value of sample i and n = number of samples

Evaluation metric	Formula	Papers	Total
Recall	$\frac{TP}{TP+FN}$	[31, 35, 36, 38, 39, 42, 45, 48, 51, 57, 58, 60, 78]	13
False positive rate (FPR)	$\frac{FP}{TN+FP}$	[31, 33, 38–40, 44, 47, 54, 57–59, 71]	12
False negative rate (FNR)	$\frac{FN}{TP+FN}$	[40, 44, 47, 48, 54, 71]	6
Precision	$\frac{TP}{TP+FP}$	[35, 36, 38, 51, 78]	5
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	[37, 48, 51, 79]	4
F-score	$2 \times \frac{precision \times recall}{precision+recall}$	[35, 36]	2
Matthew's correlation coefficient (MCC)	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$	[65]	1
Regression metrics			
Root mean squared error (RMSE)	$\sqrt{MSE}$	[46, 62, 64, 66]	4
Mean squared error (MSE)	$\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2$	[72, 76]	2
Mean absolute error (MAE)	$\frac{1}{n}\sum_{i=1}^{n}\lvert x_i - \hat{x}_i\rvert$	[61, 67]	2
Mean relative error (MRE)	$\frac{1}{n}\sum_{i=1}^{n}\frac{\lvert x_i - \hat{x}_i\rvert}{x_i}$	[30, 67]	2

mean of the available group, *available_mean* is approximately 6.87 whereas the mean of the non-available group, *non_available_mean* is 2.16. In order to compare the means of both groups, Fig. 6c shows the posterior distribution of the difference of means of both groups. There is a 99.9% probability that the mean citation rate of publications, which are using public datasets, is larger than the mean citation rate of publications, which are not using public datasets. This suggests that the publicly available datasets are easier to access, which leads to a higher citation rate for papers that involve publicly available datasets for method evaluation. Moreover, the ease of access for publicly available datasets allows researchers to directly test and compare their methods with other existing solutions for solving sensor data quality problems which are done on the same dataset.

Evaluation metrics

Apart from the different datasets used, various evaluation metrics are also seen in the selected literature. This is due to the different sensor data quality problems, for example, methods for detecting errors and methods for missing data imputation would have used different evaluation metrics to quantify its performance. The former uses *classification metrics* such as recall and precision and is based on the confusion matrix (Table 9) whereas the latter uses a *regression metrics* such as root mean squared error and mean absolute error that would quantify the difference in the estimated value and actual value. In the 39 papers out of the 57 papers that have quantitative measurements, Table 10 shows the different evaluation metrics used in those 39 papers.

Classification metrics

Recall, also known as sensitivity or true positive rate (TPR), is commonly used in error detection to calculate the number of correctly detected faults (TP) over all faults, which includes both faults that are correctly detected (TP) and faults that are incorrectly detected as a normal data point (FN). It is used as an indication of the method's ability to detect faults, which places more importance on false classification of normal data points, which are supposed to be faults, i.e. false negatives. For example, if fault detection is being used as a data-cleaning solution for the training dataset which will then be used for some other machine learning methods e.g. for prediction or analysis [31, 42], incorrectly labelling a fault as a normal data point (FN) might have some adverse effect on the next machine learning model. Other than that, precision is also used as a classification metric for fault detection. It is the number of correctly identified faults (TP) over the total faults identified, which includes both faults that are correctly detected (TP) and incorrectly detected (FP). It shows how precise the model is, by measuring in terms of all the detected faults, how many of them are actual faults. Precision differs from recall as it places more weight on the false positives instead, which is the incorrect detection of faults. This metric might be used for example, in environmental sensing applications [38] where it penalizes incorrect fault detection as this might lead to waste of manpower, cost and time, as a technician might be sent out to the deployed sensor to test and calibrate the sensor device.

The False Positive Rate (FPR), also known as the Type I error rate, is the probability of a false alarm. It is the ratio of incorrectly labelling a normal data point as a fault (FP), over all normal data points, either correctly or incorrectly labelled (TN, FP). This metric is used when Type I errors (FP) should be given higher weights. For example, in [59], FPR is used in evaluating a method for fault detection of a continuous glucose monitoring device used an artificial pancreas system. A continuous glucose monitor should not raise too many false alarms (FP) as it might cause a panic, or increase the level of distrust towards the device. False Negative Rate (FNR), on the other hand, is known as a Type II error rate or miss rate and it is the number of incorrectly labelled faults (FN) over all faults, either correctly or incorrectly labelled (TP, FN). For applications which penalize Type II errors (FN), this metric is used to evaluate the proposed method. In industrial power plants, faults that are incorrectly classified as normal data points (FN) might be detrimental to the system, as it might lead to a complete system failure. Thus, papers such as [44] have used FNR as a performance metric for their method, along with FPR.

Other performance metrics for fault detection includes accuracy, F-score, and Matthew's correlation coefficient. The accuracy takes into account all four categories of the confusion matrix: true positives, true negatives, false positives, and false negatives. However, for imbalanced datasets where the class distribution is uneven, the accuracy metric is not an ideal performance measure of a model. In sensor data, there might be more normal data points than anomalous one, contributing to the true negatives, thus making the accuracy metrics unfair for performance evaluation. F-score, on the other hand, is a function of precision and recall which balances between the two and does not take into account the number of true negatives. However, this is also a down-side to the F-score, as not including the true negatives in the calculation might give a misleading result. Moreover, both of the metrics i.e. accuracy and F-score, do not take into account the proportion of each category in the confusion matrix.

To solve this, Matthew's correlation coefficient (MCC) is a metric that correctly considers the ratio of the size of all four confusion matrix categories, allowing a higher score only if the model does well on both positive and negative categories [106]. It ranges from -1 to 1, which indicates perfect disagreement and agreement between the prediction and actual class respectively. An MCC score of 0 indicates a by chance result, which could be achieved by simply guessing that there are not any faults at all. Based on the example discussed by Chicco [106], assume that a classifier is trained on a heavily imbalanced dataset with 95 normal data and 5 anomalous data. Let the normal data be the positive class and the anomalous data be the negative class. Thus, TP $=$ correct detection of normal data, FP $=$ incorrect detection of fault as normal data, TN $=$ correct detection of fault and FN $=$ incorrect detection of normal data as fault. Suppose a model that randomly guesses all data points as normal is built. Thus, it classifies all points as positive and we have $TP = 95$, $FP = 5$, $TN = 0$ and $FN = 0$. Following the formulas for accuracy and F-score in Table 10, we have $Accuracy = 95\%$ and $F - score = 97.44\%$. However, this random guessing will be detected by MCC as it will be undefined (since the denominator will return 0), giving an indication that the classifier is not working as intended, opposed to the accuracy and F-score, which gave a false illusion that the classifier is doing well. In another example, suppose now the classifier does classify some points as faults, where $TP = 90$, $FP = 4$, $TN = 1$ and $FN = 5$, but it poorly classifies the faults as it only correctly detects 1 out of 5 faults. The accuracy and F-score are still high, resulting in $Accuracy = 91\%$ and $F - score = 95.24\%$. However, the MCC has a value of $MCC = 0.14$, which shows that it is performing poorly and there is a low correlation between the predicted class and the actual class. Thus, MCC is evidently more robust than the other two metrics and should be used more frequently to quantify fault detection method performance.

Regression metrics

These metrics are used in order to quantify the performance of methods for fault correction or missing data imputation. These metrics include the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Relative Error (MRE). MSE measures the average squared errors of the predicted values compared to the true values. However, squaring the error gives more weight to large errors. It is more useful when large errors are particularly unacceptable, but might underestimate the model's accuracy, because one large error might increase the MSE significantly. RSME is the square root of MSE, which has the same units as the quantity plotted on the vertical axis, making it more interpretable. Another metric used to evaluate the prediction of the model is the MAE and MRE. MAE measures the average of the absolute errors between the predicted values and true values. It is less sensitive to huge differences, unlike MSE or RMSE, as it takes the absolute value, not the square, of the errors.

MRE is similar to MAE, however, every data point is divided by its true value. Thus, it indicates how large the absolute error is with respect to the size of the actual data point. However, the MRE is problematic for sensor data, for which the true measurement value might be zero.

Conclusion

This paper presents the results of a systematic review of sensor data quality problems. It aims to answer the following research questions: what are the different types of errors in sensor data, how to quantify or detect and correct those errors and what domains are the different types of methods proposed in. The initial search process resulted in 13,057 publications, and by refining the search string through topic modelling, the final search string returned 6970 publications. Through the screening of these 6970 publications, 57 papers have been selected for data extraction and synthesis. The analysed publications discuss sensor data quality problems that are caused by errors in sensor data such as missing data, uncertainty, and faults, which include outliers, bias, constant values, stuck-at-zeros, and noise.

Results also show that there is a huge variety of methods suggested to detect or quantify those errors, as well as to correct them. There are 16 different types of methods presented for error detection, which are obtained from 32 papers out of the 57 selected papers that introduced techniques for the respective problem. The two most common approaches are principal component analysis (PCA) and artificial neural networks (ANN). They are both used to model the normal sensor behaviour and the newly observed readings will be compared to the model to determine if it is anomalous. Other techniques for fault detection include Ensemble Classifiers, Support Vector Machines, Clustering, and hybrid methods.

For error correction, there are ten publications that proposed methods for missing data imputation and noise correction. The most common missing data imputation technique is Association Rule Mining, with half of the respective papers proposing variations of that approach. Other approaches comprise of k-Nearest Neighbor, clustering, tensor-based singular value decomposition, and Probabilistic Matrix Factorization (PMF). On the other hand, 15 publications simultaneously address error detection and correction problems, usually termed as Fault Detection, Isolation, and Recovery (FDIR). The PCA-based approach is the most common technique for FDIR, though there are other approaches such as ANN, Bayesian Network, and hybrid methods involving Kalman filter and Dempster–Shafer theory with Ontology.

However, through this systematic review, there are several challenges that are found in this research area. From the two subsections, "Datasets and error imputation and labelling" and "Evaluation metrics", it is seen that methods from the selected literature were evaluated on different datasets, along with different pre-processing conditions and fault injection processes. The availability and ease of access of the datasets also play an essential part in helping researchers compare and evaluate their methods with other existing techniques for a particular sensor data quality problem. The Bayesian analysis of citation rates done on the 57 selected papers shows the effects of using publicly available datasets for method evaluation. There is a 99.9% probability that papers that use publicly available datasets have a higher citation rate than those that used datasets that are not publicly available, which suggests that more people are able to cite and compare their methods with those papers due to their availability online and easy access.

However, about 68% of the datasets used for evaluation are not publicly available nor reproducible. Even for the remaining 23 datasets that are used from nine publicly available sources, the data pre-processing and the error introduction step, whether by manual labelling or simulating faults artificially, are done differently. Furthermore, even for the same problem domain e.g. fault detection, fault correction, or missing data imputation, different classification and regression evaluation metrics are being used to produce a quantifiable performance measure. This provides no formal way of comparing the methods. Other than that, the use of Matthew's correlation coefficient is also shown to be more robust towards imbalanced datasets and optimistic misinterpretations. However, only one paper from the 57 selected papers is seen to have used that performance metric.

Both challenges pose a problem for this research area as they make it more difficult for researchers to compare their proposed methods with existing techniques, which may lead to counterproductive results. These two challenges show the need for an open source benchmarking system for techniques that solve sensor data quality problems. The benchmark should provide datasets complete with all the different types of errors (that is either labelled or injected artificially) and a proper scoring system that uses the appropriate evaluation metrics to allow comparability of methods in terms of their performance to solve sensor data quality issues.

Abbreviations

IoT: Internet of Things; WSNs: wireless sensor networks; PCA: principal component analysis; PRISMA: Preferred Reporting Items for Systematic Reviews and Meta-Analyses; RQ: research question; LDA: Latent Dirichlet Allocation; IC: inclusion criteria; EC: exclusion criteria; QC: quality criteria; ISO: International Standardization Organization; DQ: data quality; SVI: Sensor Validity Index; ANN: artificial neural network; TDNN: time-delay neural network; AANN: auto-associative neural network; HTM: hierarchical temporal memory; NAB: numenta anomaly benchmark; SVM: support vector machine; AGO: accumulate generating operation; ELM: extreme learning machines; SLFN: single hidden layer feedforward neural network; FARM: freshness association rule mining; t-SVD: tensor singular value decomposition; PMF: probabilistic matrix factorization; IMF: intrinsic mode functions; ARIMA: autoregressive integrated moving average; ORKF: Outlier Robust Kalman filter; LW-PLS: locally-weighted partial least squares; HVAC: heating, ventilation, and air conditioning; GSB: Grand St. Bernard; LUCE: Lausanne Urban Canopy Experiment; UCI: University of California Irvine; NAMOS: networked aquatic microbial observing system; PeMS: Performance Measurement System; TasMAN: Tasmania Marine Analysis Network; TP: true positive; TN: true negative; FP: false positive; FN: false negative; TPR: true positive rate; FPR: false positive rate; MCC: Matthew's correlation coefficient; MSE: mean squared error; RMSE: root mean squared error; MAE: mean absolute error; MRE: mean relative error.

Acknowledgements

The article processing charge was funded by the German Research Foundation (DFG) and the University of Freiburg in the funding programme Open Access Publishing.

Authors' contributions

HYT conducted the systematic review which includes gathering and extracting data from all the papers from various databases that were used for the manuscript and wrote the first revision of the manuscript. AKL developed the data analysis model. KIW proposed the systematic review topic and research questions. KIW and AKL provided direction for the literature-based review, structuring of the review, and revision of the manuscript. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

All papers analysed in this systematic review are available in ACM Digital Library, IEEE Xplore and ScienceDirect. All datasets mentioned are publicly available and their links can be found as cited.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Department of Electrical, Computer, and Software Engineering, The University of Auckland, Auckland, New Zealand. [2] Freiburg Materials Research Center, University of Freiburg, Freiburg, Germany. [3] Department of Engineering Science, The University of Auckland, Auckland, New Zealand.

Received: 20 September 2019 Accepted: 12 January 2020

Published online: 11 February 2020

References

1. Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of Things (IoT): a vision, architectural elements, and future directions. Future Gener Comput Syst. 2013;29(7):1645–60. https://doi.org/10.1016/j.future.2013.01.010.
2. Cisco: Cisco global cloud index: Forecast and methodology, 2016-2021. Whitepaper c11-738085, Cisco Systems Inc., San Jose, CA (2018). https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.pdf
3. Zhang P. Advanced industrial control technology. Oxford: William Andrew Publishing; 2010. https://doi.org/10.1016/B978-1-4377-7807-6.10003-8.
4. Wang RY, Strong DM. Beyond accuracy: what data quality means to data consumers. J Manag Inform Syst. 1996;12(4):5–33.
5. Karkouch A, Mousannif H, Al Moatassime H, Noel T. Data quality in internet of things: a state-of-the-art survey. J Netw Comput Appl. 2016;73:57–81. https://doi.org/10.1016/j.jnca.2016.08.002.
6. Christ M, Krumeich J, Kempa-Liehr AW. Integrating predictive analytics into complex event processing by using conditional density estimations. In: IEEE 20th international enterprise distributed object computing workshop (EDOCW). In: IEEE computer society, Los Alamitos, CA, USA; 2016. pp. 1–8. https://doi.org/10.1109/EDOCW.2016.7584363.
7. Zhang H, Liu J, Pang A-C. A Bayesian network model for data losses and faults in medical body sensor networks. Comput Netw. 2018;143:166–75. https://doi.org/10.1016/j.comnet.2018.07.009.
8. Ye J, Stevenson G, Dobson S. Detecting abnormal events on binary sensors in smart home environments. Pervasive Mobile Comput. 2016;33:32–49. https://doi.org/10.1016/j.pmcj.2016.06.012.
9. Li Y, Parker LE. Nearest neighbor imputation using spatial-temporal correlations in wireless sensor networks. Inform Fusion. 2014;15:64–79. https://doi.org/10.1016/j.inffus.2012.08.007.
10. Cheng R, Chen J, Xie X. Cleaning uncertain data with quality guarantees. Proc VLDB Endow. 2008;1(1):722–35. https://doi.org/10.14778/1453856.1453935.
11. Ray PP. A survey on Internet of Things architectures. J King Saud Univ Comput Inform Sci. 2018;30(3):291–319.
12. Lin J, Yu W, Zhang N, Yang X, Zhang H, Zhao W. A Survey on Internet of Things: architecture, enabling technologies, security and privacy, and applications. IEEE Intern Things J. 2017;4(5):1125–42. https://doi.org/10.1109/JIOT.2017.2683200.
13. Ahmed E, Yaqoob I, Hashem IAT, Khan I, Ahmed AIA, Imran M, Vasilakos AV. The role of big data analytics in Internet of Things. Comput Netw. 2017;129:459–71. https://doi.org/10.1016/j.comnet.2017.06.013.
14. Li Y, Chen J, Feng L. Dealing with uncertainty: a survey of theories and practices. IEEE Trans Knowl Data Eng. 2013;25(11):2463–82. https://doi.org/10.1109/TKDE.2012.179.
15. Prathiba B, Sankar KJ, Sumalatha V. Enhancing the data quality in wireless sensor networks - a review. In: 2016 international conference on automatic control and dynamic optimization techniques (ICACDOT). 2016;448–454. https://doi.org/10.1109/ICACDOT.2016.7877626.
16. Kofod-Petersen A. How to do a structured literature review in computer science. (2015).
17. Silva R, Neiva F. Systematic literature review in computer science—a practical guide. (2016). https://doi.org/10.13140/RG.2.2.35453.87524.
18. PRISMA: PRISMA—transparent reporting of systematic reviews and meta-analyses (2015). http://www.prisma-statement.org/ Accessed 08 Jan 2019.
19. Blei DM, Lafferty JD. Topic models. In: Ashok N, Srivastava MS, editors. Text mining. Classification, clustering, and applications. Chapman and Hall/CRC: New York; 2009. p. 71–93.
20. Zhai C. Statistical language models for information retrieval. Synth Lectures Human Lang Technol. 2008;1(1):1–41.
21. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12:2825–30.
22. Chow LS, Paramesran R. Review of medical image quality assessment. Biomed Sign Process Contr. 2016;27:145–54. https://doi.org/10.1016/j.bspc.2016.02.006.
23. Lapini A, Argenti F, Piva A, Bencini L. Comparison of super-resolution methods for quality enhancement of digital biomedical images. In: 2014 8th International symposium on medical information and communication technology (ISMICT). 2014. https://doi.org/10.1109/ISMICT.2014.6825243. pp. 1–5.
24. Sharma P, Sharma S. An analysis of vision based techniques for quality assessment and enhancement of camera captured document images. In: 2016 6th international conference—cloud system and Big Data engineering (Confluence). 2016. pp. 425–28. https://doi.org/10.1109/CONFLUENCE.2016.7508157.

25. Bamgboye O, Liu X, Cruickshank P. Towards modelling and reasoning about uncertain data of sensor measurements for decision support in smart spaces. In: 2018 IEEE 42nd annual computer software and applications conference (COMPSAC), 2018. pp. 744–49. https://doi.org/10.1109/COMPSAC.2018.10330.
26. Kuka C, Nicklas D. Enriching sensor data processing with quality semantics. In: 2014 IEEE international conference on pervasive computing and communication workshops (PERCOM WORKSHOPS). 2014. pp. 437–42. https://doi.org/10.1109/PerComW.2014.6815246.
27. Dunia R, Joe Qin S, Edgar TF, McAvoy TJ. Use of principal component analysis for sensor fault identification. Comput Chem Eng. 1996;20:713–8. https://doi.org/10.1016/0098-1354(96)00128-7.
28. Moher D, Liberati A, Tetzlaff J, Altman DG, Group TP. Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. PLoS Med. 2009;6(7):1–6. https://doi.org/10.1371/journal.pmed.1000097.
29. Joint Committee Guides Metrology: evaluation of measurement data-guide to the expression of uncertainty in measurement (GUM 2008). 2008.
30. Chen Y, Jiang S, Yang J, Song K, Wang Q. Grey bootstrap method for data validation and dynamic uncertainty estimation of self-validating multifunctional sensors. Chemometr Intell Lab Syst. 2015;146:63–76. https://doi.org/10.1016/j.chemolab.2015.05.003.
31. Feng J, Hajizadeh I, Samadi S, Sevil M, Hobbs N, Brandt R, Lazaro C, Maloney Z, Yu X, Littlejohn E, Quinn L, Cinar A. Hybrid online multi-sensor error detection and functional redundancy for artificial pancreas control systems. IFAC-PapersOnLine. 2018;51(18):138–43. https://doi.org/10.1016/j.ifacol.2018.09.289.
32. Harkat MF, Mourot G, Ragot J. Sensor failure detection of air quality monitoring network. IFAC Proc Vol. 2000;33(11):529–34. https://doi.org/10.1016/S1474-6670(17)37413-X.
33. Abuaitah GR, Wang B. Data-centric anomalies in sensor network deployments: analysis and detection. In: 2012 IEEE 9th international conference on mobile Ad-Hoc and sensor systems (MASS 2012), vol. Supplement. 2012. pp. 1–6. https://doi.org/10.1109/MASS.2012.6708514.
34. Ahmad S, Lavin A, Purdy S, Agha Z. Unsupervised real-time anomaly detection for streaming data. Neurocomputing. 2017;262:134–47. https://doi.org/10.1016/j.neucom.2017.04.070.
35. Bosman HHWJ, Iacca G, Tejada A, Wörtche HJ, Liotta A. Ensembles of incremental learners to detect anomalies in ad hoc sensor networks. Ad Hoc Netw. 2015;35:14–36. https://doi.org/10.1016/j.adhoc.2015.07.013.
36. Bosman HH, Iacca G, Tejada A, Wörtche HJ, Liotta A. Spatial anomaly detection in sensor networks using neighborhood information. Inform Fusion. 2017;33:41–56. https://doi.org/10.1016/j.inffus.2016.04.007.
37. Curiac D-I, Volosencu C. Ensemble based sensing anomaly detection in wireless sensor networks. Exp Syst Appl. 2012;39(10):9087–96. https://doi.org/10.1016/j.eswa.2012.02.036.
38. Dereszynski EW, Dietterich TG. Spatiotemporal models for data-anomaly detection in dynamic environmental monitoring campaigns. ACM Trans Sen Netw. 2011;8(1):3–1336. https://doi.org/10.1145/1993042.1993045.
39. Fawzy A, Mokhtar HMO, Hegazy O. Outliers detection and classification in wireless sensor networks. Egypt Inform J. 2013;14(2):157–64. https://doi.org/10.1016/j.eij.2013.06.001.
40. Hill DJ, Minsker BS. Anomaly detection in streaming environmental sensor data: a data-driven modeling approach. Environ Model Softw. 2010;25(9):1014–22. https://doi.org/10.1016/j.envsoft.2009.08.010.
41. Hou Z, Lian Z, Yao Y, Yuan X. Data mining based sensor fault diagnosis and validation for building air conditioning system. Energy Convers Manag. 2006;47(15):2479–90. https://doi.org/10.1016/j.enconman.2005.11.010.
42. Hu Y, Chen H, Li G, Li H, Xu R, Li J. A statistical training data cleaning strategy for the PCA-based chiller sensor fault detection, diagnosis and data reconstruction method. Energy Build. 2016;112:270–8. https://doi.org/10.1016/j.enbuild.2015.11.066.
43. Huang X-h. Sensor fault diagnosis and reconstruction of engine control system based on autoassociative neural network. Chin J Aeronaut. 2004;17(1):23–7. https://doi.org/10.1016/S1000-9361(11)60198-2.
44. Ibarguengoytia PH, Sucar LE, Vadera S. Real time intelligent sensor validation. IEEE Trans Power Syst. 2001;16(4):770–5. https://doi.org/10.1109/59.962425.
45. Liu H, Chen J, Huang F, Li H. An electric power sensor data oriented data cleaning solution. In: 2017 14th international symposium on pervasive systems, algorithms and networks 2017 11th international conference on frontier of computer science and technology 2017 Third international symposium of creative computing (ISPAN-FCST-ISCC). 2017. pp. 430–5. https://doi.org/10.1109/ISPAN-FCST-ISCC.2017.29.
46. Liu Y, Chen J, Sun Z, Li Y, Huang D. A probabilistic self-validating soft-sensor with application to wastewater treatment. Comput Chem Eng. 2014;71:263–80. https://doi.org/10.1016/j.compchemeng.2014.08.008.
47. Mansouri M, Harkat M-F, Nounou M, Nounou H. Midpoint-radii principal component analysis—based EWMA and application to air quality monitoring network. Chemometr Intell Lab Syst. 2018;175:55–64. https://doi.org/10.1016/j.chemolab.2018.01.016.
48. Rassam MA, Maarof MA, Zainal A. Adaptive and online data anomaly detection for wireless sensor systems. Knowl Syst. 2014;60:44–57. https://doi.org/10.1016/j.knosys.2014.01.003.
49. Sallans B, Bruckner D, Russ G. Statistical model-based sensor diagnostics for automation systems. In: Chávez, M.L., ed. Fieldbus systems and their applications Elsevier: Oxford; 2006. pp. 239–46.https://doi.org/10.1016/B978-008045364-4/50073-3. http://www.sciencedirect.com/science/article/pii/B9780080453644500733.
50. Sharifi R, Langari R. Nonlinear sensor fault diagnosis using mixture of probabilistic PCA models. Mech Syst Sign Process. 2017;85:638–50. https://doi.org/10.1016/j.ymssp.2016.08.028.
51. Solomakhina N, Hubauer T, Lamparter S, Roshchin M, Grimm S. Extending statistical data quality improvement with explicit domain models. In: 2014 12th IEEE international conference on industrial informatics (INDIN). 2014. pp. 720–5. https://doi.org/10.1109/INDIN.2014.6945602.
52. Tsang KM. Sensor data validation using gray models. ISA Trans. 2003;42(1):9–17. https://doi.org/10.1016/S0019-0578(07)60109-8.
53. Tsang KM, Chan WL. Data validation of intelligent sensor using predictive filters and fuzzy logic. Sens Actuat A. 2010;159(2):149–56. https://doi.org/10.1016/j.sna.2010.03.013.

54. Xiao H, Huang D, Pan Y, Liu Y, Song K. Fault diagnosis and prognosis of wastewater processes with incomplete data by the auto-associative neural networks and ARMA model. Chemometr Intell Lab Syst. 2017;161:96–107. https://doi.org/10.1016/j.chemolab.2016.12.009.
55. Liu Y, Daoping H, Zhifu L. A SEVA soft sensor method based on self-calibration model and uncertainty description algorithm. Chemometr Intell Lab Syst. 2013;126:38–49. https://doi.org/10.1016/j.chemolab.2013.04.009.
56. Yu Z, Bedig A, Montalto F, Quigley M. Automated detection of unusual soil moisture probe response patterns with association rule learning. Environ Modell Softw. 2018;105:257–69. https://doi.org/10.1016/j.envsoft.2018.04.001.
57. Zhang Y, Meratnia N, Havinga P. Adaptive and online one-class support vector machine-based outlier detection techniques for wireless sensor networks. In: 2009 international conference on advanced information networking and applications workshops. 2009. pp. 990–5. https://doi.org/10.1109/WAINA.2009.200.
58. Zhang Y, Meratnia N, Havinga PJM. Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine. Ad Hoc Netw. 2013;11(3):1062–74. https://doi.org/10.1016/j.adhoc.2012.11.001.
59. Zhao C, Fu Y. Statistical analysis based online sensor failure detection for continuous glucose monitoring in type I diabetes. Chemometr Intell Lab Syst. 2015;144:128–37. https://doi.org/10.1016/j.chemolab.2015.04.001.
60. Yang J, Lin L, Sun Z, Chen Y, Jiang S. Data validation of multifunctional sensors using independent and related variables. Sens Actuat A. 2017;263:76–90. https://doi.org/10.1016/j.sna.2017.05.015.
61. Chok H, Gruenwald L. Spatio-temporal association rule mining framework for real-time sensor network applications. In: Proceedings of the 18th ACM conference on information and knowledge management. CIKM '09. ACM: New York; 2009. pp. 1761–4. https://doi.org/10.1145/1645953.1646224. Accessed 31 Aug 2018.
62. D'Aniello G, Gaeta M, Hong TP. Effective quality-aware sensor data management. IEEE Trans Emerg Top Comput Intell. 2018;2(1):65–77. https://doi.org/10.1109/TETCI.2017.2782800.
63. Fekade B, Maksymyuk T, Kyryk M, Jo M. Probabilistic recovery of incomplete sensed data in IoT. IEEE Intern Things J. 2017;. https://doi.org/10.1109/JIOT.2017.2730360.
64. Gruenwald L, Chok H, Aboukhamis M. Using data mining to estimate missing sensor data. In: Seventh IEEE international conference on data mining workshops (ICDMW 2007), 2007. pp. 207–12. https://doi.org/10.1109/ICDMW.2007.103.
65. Tang J, Zhang G, Wang Y, Wang H, Liu F. A hybrid approach to integrate fuzzy C-means based imputation method with genetic algorithm for missing traffic volume data estimation. Transport Res C. 2015;51:29–40. https://doi.org/10.1016/j.trc.2014.11.003.
66. Wang Y, Wang J, Li H. An interpolation approach for missing context data based on the time-space relationship and association rule mining. In: 2011 third international conference on multimedia information networking and security, 2011. pp. 623–7. https://doi.org/10.1109/MINES.2011.78.
67. Xu P, Ruan W, Sheng QZ, Gu T, Yao L. Interpolating the missing values for multi-dimensional spatial-temporal sensor data: a tensor SVD approach. In: Proceedings of the 14th EAI international conference on mobile and ubiquitous systems: computing, networking and services. MobiQuitous 2017. pp. 442–51. ACM: New York; 2017. https://doi.org/10.1145/3144457.3144474.
68. Hermans F, Dziengel N, Schiller J. Quality estimation based data fusion in wireless sensor networks. In: 2009 IEEE 6th international conference on mobile adhoc and sensor systems. 2009. pp. 1068–70. https://doi.org/10.1109/MOBHOC.2009.5337006.
69. Alawi A, Choi SW, Martin E, Morris J. Sensor fault identification using weighted combined contribution plots. In: Zhang H-Y, ed. Fault detection, supervision and safety of technical processes 2006. 2007. pp. 908–13. https://doi.org/10.1016/B978-008044485-7/50153-6. http://www.sciencedirect.com/science/article/pii/B9780080444857501536.
70. Smarsly K, Law KH. Decentralized fault detection and isolation in wireless structural health monitoring systems using analytical redundancy. Adv Eng Softw. 2014;73:1–10. https://doi.org/10.1016/j.advengsoft.2014.02.005.
71. Tadić P, Đurović Z. Particle filtering for sensor fault diagnosis and identification in nonlinear plants. J Process Control. 2014;24(4):401–9. https://doi.org/10.1016/j.jprocont.2014.02.009.
72. Uren KR, Schoor Gv, Rand CPd, Botha A. An integrated approach to sensor FDI and signal reconstruction in HTGRs—Part I: theoretical framework. Ann Nucl Energy. 2016;87:750–60. https://doi.org/10.1016/j.anucene.2015.06.010.
73. Yu Y, Li H. Virtual in-situ calibration method in building systems. Autom Constr. 2015;59:59–67. https://doi.org/10.1016/j.autcon.2015.08.003.
74. Wang Y, Yang A, Li Z, Wang P, Yang H. Blind drift calibration of sensor networks using signal space projection and Kalman filter. In: 2015 IEEE tenth international conference on intelligent sensors, sensor networks and information processing (ISSNIP). 2015. pp. 1–6. https://doi.org/10.1109/ISSNIP.2015.7106904.
75. Zahedi S, Szczodrak M, Ji P, Mylaraswamy D, Srivastava M, Young R. Tiered architecture for on-line detection, isolation and repair of faults in wireless sensor networks. In: MILCOM 2008–2008 In: IEEE military communications conference. 2008. pp. 1–7. https://doi.org/10.1109/MILCOM.2008.4753634.
76. Omitaomu OA, Protopopescu VA, Ganguly AR. Empirical mode decomposition technique with conditional mutual information for denoising operational sensor data. IEEE Sens J. 2011;11(10):2565–75. https://doi.org/10.1109/JSEN.2011.2142302.
77. Sadıkoglu F, Kavalcıoğlu C. Filtering continuous glucose monitoring signal using Savitzky–Golay filter and simple multivariate thresholding. Proc Comput Sci. 2016;102:342–50. https://doi.org/10.1016/j.procs.2016.09.410.
78. Jäger G, Zug S, Brade T, Dietrich A, Steup C, Moewes C, Cretu AM. Assessing neural networks for sensor fault detection. In: 2014 IEEE international conference on computational intelligence and virtual environments for measurement systems and applications (CIVEMSA). 2014. pp. 70–5. https://doi.org/10.1109/CIVEMSA.2014.6841441.
79. Rahman A, Smith DV, Timms G. A novel machine learning approach toward quality assessment of sensor data. IEEE Sens J. 2014;14(4):1035–47. https://doi.org/10.1109/JSEN.2013.2291855.
80. Richter C. Reliability assessment in everyday-objects based physical-activity sensing using personal information. In: Proceedings of the 8th ACM international conference on pervasive technologies related to assistive environments. PETRA '15, pp. 39–1394. ACM: New York; 2015. https://doi.org/10.1145/2769493.2769548.

81. Wang P, Gao RX, Tang X, Fan Z. Sensing uncertainty evaluation for product quality. Proc CIRP. 2016;41:706–11. https://doi.org/10.1016/j.procir.2015.12.105.
82. Aggarwal CC. An introduction to outlier analysis. Outlier analysis. Springer: New York; 2013. p. 1–40. https://doi.org/10.1007/978-1-4614-6396-2_1.
83. Ahmad NF, Hoang DB, Phung MH. Robust preprocessing for health care monitoring framework. In: 2009 11th international conference on e-Health networking, applications and services (Healthcom). 2009. pp. 169–74. https://doi.org/10.1109/HEALTH.2009.5406196.
84. Rabatel J, Bringay S, Poncelet P. Anomaly detection in monitoring sensor data for preventive maintenance. Expert Syst Appl. 2011;38(6):7003–15. https://doi.org/10.1016/j.eswa.2010.12.014.
85. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical recipes. The art of scientific computing. 3rd ed. Cambridge: Cambridge University Press; 2007.
86. Kramer MA. Autoassociative neural networks. Comput Chem Eng. 1992;16(4):313–28. https://doi.org/10.1016/0098-1354(92)80051-A.
87. Hawkins J, Blakeslee S. On intelligence. New York: Times Books; 2004.
88. Numenta: Numenta—Home of the HTM Community (2019). https://numenta.org/. Accessed 08 Jan 2019.
89. Fisher RA. Statistical methods for research workers. In: Kotz S, Johnson NL, editors. Breakthroughs in statistics: methodology and distribution Springer series in statistics. Springer: New York; 1992. p. 66–70. https://doi.org/10.1007/978-1-4612-4380-9_6.
90. Christ M, Braun N, Neuffer J, Kempa-Liehr AW. Time series featuRe extraction on basis of scalable hypothesis tests (tsfresh—a python package). Neurocomputing. 2018;307:72–7. https://doi.org/10.1016/j.neucom.2018.03.067.
91. Deng J-L. Control problems of grey systems. Syst Contr Lett. 1982;1(5):288–94. https://doi.org/10.1016/S0167-6911(82)80025-X.
92. Huang G-B, Zhu Q-Y, Siew C. Extreme learning machine: a new learning scheme of feedforward neural networks. Neural Netw. 2004;2:985–9902. https://doi.org/10.1109/IJCNN.2004.1380068.
93. Ingelrest F, Barrenetxea G, Schaefer G, Vetterli M, Couach O, Parlange M. Sensorscope: application-specific sensor network for environmental monitoring. ACM Trans Sens Netw. 2010;6(2):17.
94. Barrenetxea G. Sensorscope: Sensor Networks for Environmental Monitoring (2018). https://doi.org/10.5281/zenodo.2654726. https://lcav.epfl.ch/research/research-archives/research-archives-communications_and_sensor_networks_archive-html/sensorscope-en/page-145180-en-html/. Accessed 08 May 2019.
95. Madden S. Intel Lab Data (2004). http://db.csail.mit.edu/labdata/labdata.html. Accessed 08 May 2019.
96. Dua D, Graff C. UCI machine learning repository (2017). http://archive.ics.uci.edu/ml Accessed 08 May 2019.
97. University of Southern California: Networked Aquatic Microbial Observing System (NAMOS). http://robotics.usc.edu/~namos/data.html. 2002.
98. Numenta: the numenta anomaly benchmark. 2019. https://github.com/numenta/NAB. Accessed 08 May 2019.
99. of California S. California department of transportation: caltrans performance measurement system; 2019. http://pems.dot.ca.gov/. Accessed 08 May 2019.
100. Timms G, Sharman C, Howell B, McCulloch J, Hugo D. Tasmanian marine analysis network—Sullivans Cove CSIRO Wharf Sensor. 2012;. https://doi.org/10.4225/08/50613AE767787. https://data.csiro.au/collections/#collection/CIcsiro:5604v1. Accessed 08 May 2019.
101. Wren CR, Ivanov YA, Leigh D, Westhues J. The merl motion detector dataset. In: Workshop on massive datasets (MD). 2007. pp. 10–14. http://www.merl.com/publications/TR2007-069.
102. Wren C, Ivanov Y. MERLSense Data (2009). https://sites.google.com/a/drwren.com/wmd/home. Accessed 08 May 2019.
103. PhysioNet: PhysioNet: the research resource for complex physiologic signals (2019). https://physionet.org/. Accessed 08 May 2019.
104. Kruschke J. Bayesian estimation supersedes the t test. J Exp Psychol Gen. 2012;. https://doi.org/10.1037/a0029146.
105. Salvatier J, V Wiecki T, Fonnesbeck C. Probabilistic programming in python using pymc3. 2016. https://doi.org/10.7287/PEERJ.PREPRINTS.1686V1.
106. Chicco D. Ten quick tips for machine learning in computational biology. BioData Mining. 2017. p. 10. https://doi.org/10.1186/s13040-017-0155-3. Accessed 17 Mar 2019.

Publisher's Note