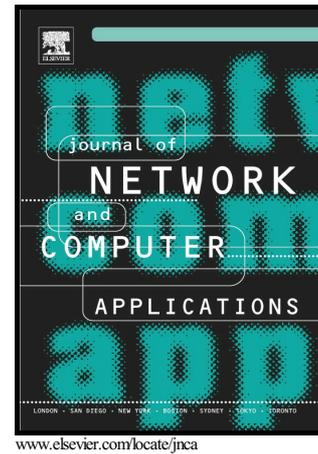


Author's Accepted Manuscript

Data Quality in Internet of Things: A state-of-the-art survey

Aimad Karkouch, Hajar Mousannif, Hassan Al Moatassime, Thomas Noel



PII: S1084-8045(16)30156-4
DOI: <http://dx.doi.org/10.1016/j.jnca.2016.08.002>
Reference: YJNCA1684

To appear in: *Journal of Network and Computer Applications*

Received date: 31 March 2016
Revised date: 12 July 2016
Accepted date: 1 August 2016

Cite this article as: Aimad Karkouch, Hajar Mousannif, Hassan Al Moatassime and Thomas Noel, Data Quality in Internet of Things: A state-of-the-art survey. *Journal of Network and Computer Applications* <http://dx.doi.org/10.1016/j.jnca.2016.08.002>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting galley proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain

Data Quality in Internet of Things: A state-of-the-art survey

Aimad Karkouch^{*,a}, Hajar Mousannif^b, Hassan Al Moatassime^a, Thomas Noel^c

^aOSER research team, Computer Science Department, FSTG, Cadi Ayyad University, Morocco

^bLISI Laboratory, Computer Science Department, FSSM, Cadi Ayyad University, Morocco

^cICube Laboratory, University of Strasbourg, France

In the Internet of Things (IoT), data gathered from a global-scale deployment of smart-things, are the base for making intelligent decisions and providing services. If data are of poor quality, decisions are likely to be unsound. Data quality (DQ) is crucial to gain user engagement and acceptance of the IoT paradigm and services. This paper aims at enhancing DQ in IoT by providing an overview of its state-of-the-art. Data properties and their new lifecycle in IoT are surveyed. The concept of DQ is defined and a set of generic and domain-specific DQ dimensions, fit for use in assessing IoT's DQ, are selected. IoT-related factors endangering the DQ and their impact on various DQ dimensions and on the overall DQ are exhaustively analyzed. DQ problems manifestations are discussed and their symptoms identified. Data outliers, as a major DQ problem manifestation, their underlying knowledge and their impact in the context of IoT and its applications are studied. Techniques for enhancing DQ are presented with a special focus on data cleaning techniques which are reviewed and compared using an extended taxonomy to outline their characteristics and their fitness for use for IoT. Finally, open challenges and possible future research directions are discussed.

Keywords: Internet of things, data quality, data cleaning, outlier detection

* Corresponding author.

Authors' e-mail addresses: aimad.karkouch@ced.uca.ac.ma (A. Karkouch); mousannif@uca.ma (H. Mousannif); hassan.al.moatassime@gmail.com (H. Al Moatassime); noel@unistra.fr (T. Noel).

1. INTRODUCTION

The Internet of Things (IoT) is about millions of connected, communicating and exchanging objects, scattered all over the world and generating tremendous amounts of data using their sensors every single second. IoT is a new evolution of the Internet (Evans, 2011) and has many definitions depending on the chosen viewpoint. One that relates to data reports the shifting of roles in the era of IoT. Interconnected smart things will become the major data producers and consumers instead of humans. The flow of data from the physical to the digital world will extend the awareness of computers of their surroundings, thus, gaining the ability to act on behalf of humans through ubiquitous services.

IoT has and will affect many fields in our daily life both on personal and business levels (e.g. cities, homes, health, etc.). Further, it has a significant impact on society to the extent it has become a social “symbolic capital of power” (Nataliia and Elena, 2015). A taxonomy of IoT applications is presented in (Gubbi et al., 2013) which, based on the type of network availability, coverage, scale, heterogeneity, repeatability, user involvement and impact, identifies four application domains: Home and personal, enterprise, utilities and mobile. Applications based on the crossing-over of physical and cyber worlds allowed by the IoT vision (e.g. Health applications, Home energy monitoring, Smart cities, Intelligent Products, etc.) have already been created and many more are expected (Aggarwal et al., 2013; Kiritsis, 2011).

Data represent the bridge that connects cyber and physical worlds. Their importance is illustrated with the emergence of IoT semantic-oriented vision (Atzori et al., 2010) which finds its utility from the need of ways to represent and manipulate the huge amount of raw data expected to be generated from the “things”. The autonomous and continuous harvesting of data by the “things” (e.g. RFID readers, sensor nodes, etc.) easily overtakes manually entered data. It was in 2008 when the number of connected objects has already surpassed the number of persons on the planet (Aggarwal et al., 2013). Moreover, considering the predictions in (National Intelligence Council, 2008; Sundmaeker et al., 2010), the number of connected objects will become even greater. In fact, as predicted in (National Intelligence Council, 2008), common things of our daily life (e.g. lamps, refrigerators, food packages, etc.) will have had embedded components allowing them to communicate and become more intelligent by the year 2025. Furthermore, technological advances have impressively sharpened the “data harvesting” capabilities of embedded sensor devices resulting in more generated data and more continuous data streams from the real world. As a result, IoT has become an important catalyzer of Big Data Analytics.

Data are a valuable asset in the IoT because they give insights about a given phenomenon, person or entity which are used by applications to provide intelligent services in a ubiquitous manner. These insights are mined from the harvested data using data mining techniques and algorithms (Tsai et al., 2014). Many works (Equille, 2007; Hand et al., 2001; Hipp et al., 2001) state the importance of data quality (DQ) for data mining processes and the impact of low DQ on the validity of the results and interpretations of such processes, leading to the conclusion that DQ and accuracy should be ensured. However, many factors characterizing the IoT including deployment scale, things’ constrained resources (Branch et al., 2009) and intermittent loss of connection (Zeng et al., 2011) are endangering the quality of the produced data. Many DQ problems, measurable at the level of DQ dimensions, occur as a result of such hazardous elements. One major manifestation of these deviations in DQ are Data Outliers (Branch et al., 2009; Chandola et al., 2009; Javed and Wolf, 2012; Otey et al., 2006). However, while outliers could describe errors, they can also describe rare events (Zhang et al., 2010) which represent precious information for the

applications (Knox and Ng, 1998) (e.g. “unusual” high temperature readings as a result of a fire in a monitored forest). Solutions to deal with DQ problems are required considering that data trustworthiness is crucial for user engagement and acceptance of IoT services, and thus, for a successful large scale deployment of the IoT paradigm. This is shown in (Yan et al., 2014) where data collection trust and accuracy represent the major concern of the Data Perception Trust as part of a holistic trust management approach for IoT.

Our survey investigates DQ in the context of IoT. First of all, we present the new lifecycle of data in IoT and we review the characteristics of data gathered by the things. Then, we enumerate IoT environment-related factors affecting the quality of data and we present their distribution. Further, we exhaustively analyze the impact of each previously-mentioned factor on various DQ dimensions, which we selected for assessing IoT data, and thus on the overall DQ. Moreover, we identify in what form do DQ problems manifest and we associate each manifestation class with its symptoms with respect to the affected DQ dimensions. Further, we study data outliers as a major type of DQ problems by defining their underlying concept, enumerating their types and analyzing their impact on a large scale deployment and acceptance of IoT paradigm. We further investigate outliers’ impact on key components of IoT applications. Techniques for enhancing DQ and overcoming DQ problems are then studied. We compare and discuss, in more depth, data cleaning techniques that promise to overcome the uncertainty in the gathered data and provide purified data for IoT applications. Finally, we discuss open challenges and possible future research directions we believe have the potential to deliver efficient solutions and approaches. Our aim is to give an overview of the current state of the art of DQ in the context of IoT in order to find ways to enhance it. Even though, some surveys (Qin et al., 2014; Sathe et al., 2013; Zhang et al., 2010) have been conducted recently, they either present the whole data processing cycle from acquisition to compression with a lack of focus on DQ management, or present data cleaning techniques for a sub-component of the IoT. In fact, (Qin et al., 2014) only surveys the data stream processing techniques, data storage, search and event processing in IoT. (Sathe et al., 2013) presents data acquisition, query processing and data compression techniques. It also surveys mathematical models used in outlier detection techniques. However, it only provides brief descriptions of actual data cleaning techniques such as the declarative-based one. Further, no comparison of these cleaning techniques is presented. (Zhang et al., 2010) presents techniques for outlier detection in Wireless Sensor Networks (WSN) which only represents a sub-component of IoT.

The remainder of this article is organized as follows. Section 2 discusses the new lifecycle of data in the context of IoT and their characteristics. Also, we introduce the concept of DQ and its dimensions. We then specify a set of generic and domain-specific DQ dimensions that are fit for assessing DQ in IoT. In Section 3, IoT-related factors endangering the DQ are enumerated and their distribution in a 3-layered IoT architecture is described. Further, an exhaustive qualitative analysis of their impact on various DQ dimensions and on the overall DQ is presented. In Section 4, we study how DQ problems manifest in IoT and what are their symptoms (i.e. affected DQ dimensions). In Section 5, we study, in more depth, the concept of data outliers as a major DQ problem. Their types and more importantly their impact on IoT is discussed. In Section 6, we study DQ enhancement techniques that promise to overcome DQ problems. We largely focus on data cleaning techniques for which we present a general architecture, followed with the specification of a comparison taxonomy and a comparison of data cleaning techniques that outlines their properties

and how fit they are for use in IoT context. Section 7 discusses open challenges and possible future research directions. Finally, Section 8 concludes the article.

2. DATA AND DQ IN IOT

Data represent a valuable asset in the IoT paradigm as a source for extracting insights and a means for communication. Moreover, quality is a critical requirement for any data consumer (e.g. IoT pervasive services and their users). In this section, we present the new data lifecycle in the context of IoT vision. We also discuss IoT data's characteristics. Furthermore, we define the concept of DQ. Then, we introduce DQ dimensions and their categories as metrics for measuring this quality aspect of data. Further, starting from various DQ dimensions used in the context of WSN and RFID-enabled data, we identify numerous DQ dimensions that could be used for assessing IoT's data. Finally, considering the scope of the IoT, we investigate other domain-specific DQ dimensions potentially usable in specific IoT application scenarios.

2.1 A new data lifecycle

As shown in Fig. 1, in the well-known conventional internet, data come generally from people using their computers (e.g. to interact with each other on social networks) and are used generally to provide services for these same people. In contrast, in IoT, things will produce the majority of data and will also be their main consumer in order to provide services for persons. Further, data are the main communication medium in the Machine-2-Machine (M2M) paradigm (Aggarwal et al., 2013), which is a predecessor of the IoT (Holler et al., 2014); a paradigm that will help objects in the IoT communicate and collaborate to autonomously provide new services. Data are a valuable asset in the IoT because they give insights about a given phenomenon, person or entity. Those insights are used by applications to provide intelligent services in a ubiquitous manner. If data are inaccurate, extracted knowledge and action based on it will probably be unsound.

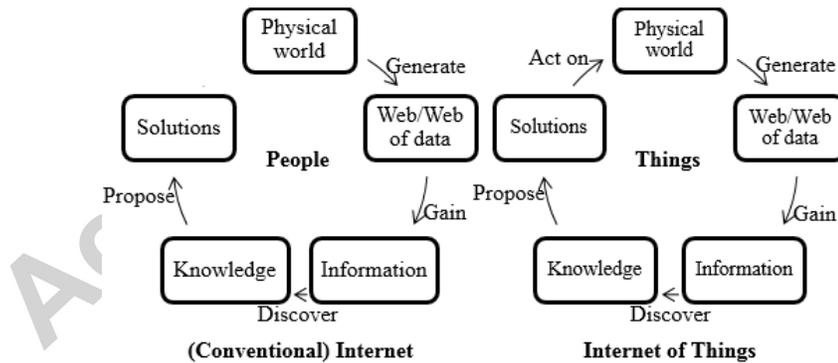


Fig. 1. Data life cycle in the Internet and IoT (Adapted from (Qin et al., 2014))

2.2 IoT data characteristics

IoT sensors generally monitor a variable of interest (e.g. temperature, sleep habits, etc.) in the physical world. Moreover, the environments in which the harvesting of data occurs are rapidly changing and volatile (Qin et al., 2014). As a result, many characteristics (Javed and Wolf, 2012; Sathe et al., 2013) are usually associated with data in the IoT. While some of these characteristics might be considered omnipresent (i.e. uncertain, erroneous, noisy, distributed and voluminous), other characteristics are not general and highly depend on the context and the monitored phenomena (i.e.

smooth variation, continuous, correlation, periodicity and Markovian behavior). Below is a summary of these characteristics:

- Uncertain, erroneous and noisy: Considering the numerous factors (Section 3) endangering the quality of data, generated data in the IoT are considered to be inherently uncertain and erroneous.
- Voluminous and distributed: In the IoT, sources of data are millions of devices scattered all over the world. The generation rate is enormous and easily overwhelms its human-generated counterpart.
- Smooth variation: Many physical monitored variables of interest (e.g. ambient temperature) exhibit smooth variations, i.e. a small variation (or none) occurs between 2 consecutive time stamps.
- Continuous: Data produced from monitoring many physical phenomenon is continuous (e.g. temperature, etc.) even when a sampling strategy is adopted. This is a direct result of the smooth variations. Sampling is used primarily to achieve energy-efficiency because the monitored phenomenon does not often change in a sudden.
- Correlation: Generated sensor value dataset has often an underlying correlation. The data are either temporally correlated, spatially correlated or both.
- Periodicity: Dataset related to many phenomenon may present an inherent periodic pattern where the same values occur at specific intervals.
- Markovian behavior: The sensor value at a given time stamp t_i is only function of the previous sensor value at the previous timestamp t_{i-1} .

2.3 Definition of DQ and DQ Dimensions

DQ refers to how well data meet the requirements of data consumers (Batini and Scannapieco, 2006; Wang and Strong, 1996). This definition gives a broader conceptualization of DQ by focusing on how consumers perceive quality, rather than the information systems professionals' perception limited to intrinsic level and accuracy dimension, considering that data have become a product, the fitness of which is judged by its user. This means that DQ would hardly be seen in the same way by different users. In fact, each data consumer requires the used data to fulfill certain criteria which he presumes essential for his own tasks at hand. These criteria or aspects or attributes of DQ are known as DQ Dimensions (e.g. Accuracy, Timeliness, Precision, Completeness, Reliability and Error recovery (Bailey and Pearson, 1983; Batini and Scannapieco, 2006; Geisler et al., 2011; Klein and Lehner, 2009b; Strong et al., 1997)). Based on this broader conceptualization and starting from 159 dimensions, four main categories have been identified (Wang and Strong, 1996) as described in Table 1.

Moreover, there exist a plethora of DQ dimensions (both domain-agnostic and domain-specific) due to the fact that data are a representation of various aspects of the real world phenomena (Batini and Scannapieco, 2006). However, there is no standardized definition for each and every dimension. In fact, a single dimension could have many (and different) definitions and could be considered with respect to various granules. We take for example the Timeliness dimension; different definitions taken from previous works in the DQ literature clearly show a non-agreement on a single definition. In fact, (Dasu and Johnson, 2003) defines timeliness as "the currency of the data. That is, the most recent time when it was updated". In (Klein and Lehner, 2009b), timeliness is seen from two perspectives; as

“the age of a specific data item as the difference between the recording timestamp and the current system time” and as “the punctuality of the data item with respect to the application context”. Moreover, in (Naumann, 2002), timeliness is seen as “the average age of the data in a source”. Finally, in (Liu and Chi, 2002), timeliness is defined as “The extent to which data are sufficiently up-to-date for a task.”

Table 1. Categories of DQ dimensions

DQ Dimensions' category	Definition	Examples
Intrinsic	Dimensions that describe quality that is innate in or that inherently exists within data.	Accuracy, Reputation
Contextual	Dimensions describing the quality with respect to the context of tasks using data.	Timelines, Completeness, Data volume
Representational	Dimensions describing how well data formats are representative and understandable.	Interpretability, Ease of understanding
Accessibility	Dimensions that describe how accessible (and in the same time secured) data are for data consumers.	Accessibility, Access security

2.4 DQ and DQ dimensions for IoT

Back in 2003, (Dasu and Johnson, 2003) asserted that contemporary data need updated and more flexible criteria for their assessment. These contemporary data were characterized by the nature of their collection process and federated aspect, size, variety and content.

Likewise, the emergence of the IoT paradigm, which takes all the characteristics of the contemporary data to a whole new level (e.g. the size of gathered data), may also require updated criteria to assess its data.

For the IoT, DQ means essentially how suitable the gathered data (from the smart things) are for providing ubiquitous services for IoT users. As the WSN and RFID are the key enabling technologies of the IoT paradigm, it makes sense to adopt DQ dimensions used for assessing WSN-enabled and RFID-enabled data to equally assess IoT data. However, we believe that DQ dimensions for IoT should cover a larger vision than those for WSN and RFID.

Klein and Lehner (2009b) have used five dimensions for assessing the quality of sensor data streams (and for data streaming environments in general) namely accuracy, confidence, completeness, data volume and timeliness. For RFID data, (van der Togt et al., 2011) proposes a framework for evaluating the performance and assessing the DQ of RFID systems especially in healthcare settings. Some of the key phases of this framework focus on the data accuracy and data completeness dimensions and their assessment. Also, (Sellitto et al., 2007) reports information quality attributes benefits resulting from the adoption of RFID technologies in the retail supply chain domain. These reported quality attributes could be projected on datasets given that pieces of information are themselves originated from data. For

example, accurate information are essentially extracted from accurate data. Even though these quality attributes were identified with respect to a specific application domain, i.e. retail supply chain, many of them are not restricted to it (e.g. accuracy, timeliness, completeness, accessibility, etc.).

In the following Table 2, we report common quality dimensions defined for WSN and RFID systems. The examples given in the table are considered in the following scenario: A set of n smart things each of which equipped with one sensor capable of measuring temperature. The sensors have a precision class of 5% and could measure up to a maximum 80 degrees Celsius. The sensors' sampling rate is set to one value/minute. Each tuple i of the data stream contains the measured temperature value v and a timestamp t_i .

Table 2. Common DQ dimensions defined for WSN and RFID – enabled data

DQ Dimension	Definition	Category	Example	Variation
Accuracy	The maximal absolute systematic error α such that the real values belong to the interval $[v - \alpha, v + \alpha]$.	Intrinsic	Considering the characteristics of the sensors described above (e.g. Precision class and measurement range), the absolute accuracy error $\alpha = 4$ °C.	Constant
Confidence	The statistical error ϵ such that $[v - \epsilon, v + \epsilon]$ contains the real value with a confidence probability of p .	Intrinsic	We consider the following dataset of values [20, 25, 23, 19.5, 18.2, 28, 22.2, 18.4 and 16.5] and a configuration as in (Klein and Lehner, 2009b). For a confidence probability of $p=99\%$, the confidence $\epsilon= 3.47$ °C	Random
Completeness	The ratio of non-interpolated items to all available (i.e. both non-interpolated and interpolated) data in the considered stream window.	Contextual	We consider a window of 2 hours. If each sensor fails twice to report during a stream window, then the completeness of this stream window is $\frac{(120-2)^n}{120n} \approx 0.983$.	Random
Data volume	The number of raw data items (values) available for use to compute a result data item (in a stream query or sub-query)	Contextual	Consider the following query: Select max (temp) from stream where timestamp \geq now-7200s. The Data Volume used to compute the result data item (The maximum temperature in the last two hours) is $120 * n$. If we also consider the completeness of this stream window to be equal to 0.983 and we suppose that missing values have not been interpolated, then the Data Volume will be $n * 120 * 0.983 \approx 118n$	Random
Timeliness	The difference between the current timestamp and the recording timestamp. May express both the age and the punctuality of a data item.	Contextual	We consider the current timestamp $t_{current}=1441985872$ and a tuple i with a timestamp $t_i=1441984000$. Timeliness (Tuple i) = $1441985872 - 1441984000 = 1872s$.	Not constant (Calculated at runtime)

Moreover, in the light of IoT paradigm, other DQ dimensions, in addition to those presented in Table 2, could be introduced to evaluate IoT DQ. In fact, with the predicted huge amount of collected data, requirements such as ease of access become essential. Also, the things forming the IoT will be scattered around the world, yet

they need to be accessed securely. In addition, IoT applications rely on heterogeneous and distributed sources of data which need to be interpretable and have concise representation for easy and meaningful integration. Further, these dimensions are tightly related to security challenges facing IoT which have been extensively surveyed in (Jing et al., 2014; Sicari et al., 2015; Zhao and Ge, 2013).

In Table 3, we report additional DQ dimensions as defined in the relational data field (Strong et al., 1997; Wang and Strong, 1996). Also, it is worth noting that in (Wang and Strong, 1996), accessibility is considered a part of the overall DQ rather than a separated field of study.

Table 3. Additional DQ dimensions for IoT

DQ dimension	Definition	Category
Ease of access	The availability and easiness of retrieving data.	Accessibility
Access security	Securing data in order to protect its privacy and confidentiality.	Accessibility
Interpretability	Data is clear in meaning and format.	Representational

It is worth noting that our set of DQ dimensions for assessing IoT DQ is larger than the one used in (Sicari et al., 2014) for their proposed IoT system architecture and its application case study. In fact, the authors only used accuracy, completeness and timeliness dimensions. Moreover, (Guo et al., 2013) considers data accuracy alongside source validity as important DQ criteria in the context of IoT applications. Furthermore, (F. Li et al., 2012) only proposed using the currency, availability and validity DQ dimensions in pervasive applications where currency and validity dimensions are closely related to timeliness and accuracy dimensions respectively.

2.5 DQ dimensions for IoT domain-specific applications

Looking at the scope of the IoT and the numerous applications that could be built using its paradigm, we assume that there could be a need for other domain-specific DQ dimensions. As we have already mentioned, DQ dimensions could be either domain-dependent or domain-independent. In Table 4, we present some domain-specific DQ dimensions defined in the context of different IoT application domains (Cardoso and Carreira, n.d.; Nobles et al., 2015; Pinto-Valverde et al., 2013).

Table 4. IoT Domain-specific DQ dimensions

DQ Dimension	Definition	Domain
Duplicates	Healthcare records duplication's rate in the patients' databases.	E-health
Availability	Data, i.e. Electronic Health Record (HER) records, which are available for a secondary use in epidemiological research.	E-health
Duplicates	Evaluates if a reading is not being received and stored more than once.	Smart grids

2.6 DQ dimensions trade-offs

It is worth noticing that trade-offs usually occur when handling certain DQ dimensions such as the pair accuracy and timeliness (Geisler et al., 2011). In fact, DQ dimensions are correlated and trade-offs could be necessary when highlighting one dimension over others (Batini and Scannapieco, 2006). For example, usually when aiming for accurate data, many checks and verifications are involved that may cause delays in data arrival (i.e. affecting timeliness). Inversely, to obtain timely data, we may need to neglect the checks pipeline which could affect the accuracy of data. This scenario could be even worse when resources are limited as in the case of smart things which, generally, cannot handle many operations at a time.

Many optimizing mechanisms designed for DQ dimensions' trade-offs have been proposed for different contexts. As an example, (Ballou and Pazer, 1995) proposes a methodology to define when the accuracy-timeliness trade-off is optimal for decision making, i.e. when the available data is accurate enough to still give a timely and sound enough decision. Also, the proposed framework could well work for other trade-offs such as completeness-timeliness ones. Moreover, (Ballou and Pazer, 2003) provides a framework to optimize the consistency-completeness trade-off enabling data consumers to determine when it is more beneficial to keep data even if it is inconsistent (i.e. better completeness) versus when to discard them (i.e. better consistency). Finally, (Helfert et al., 2009) proposes a cost/benefit model for optimizing the security-timeliness trade-offs especially for real-time applications.

3. FACTORS ENDANGERING IOT DQ AND THEIR IMPACT

In IoT context, various factors could represent potential hazardous elements to DQ. In this section, we represent these factors as well as their distribution in a 3-layered IoT architecture. Further, through an exhaustive analysis, we qualitatively study their impact on different DQ dimensions and thus on the overall DQ. As a result, a mapping between IoT factors and IoT DQ is created.

3.1 Factors affecting IoT DQ

In the context of IoT, millions of sensing-enabled devices will be deployed in various areas, regions and environments to monitor different phenomenon and produce insights based on which further actions and goals are achieved. Many problems (Branch et al., 2009; Erguler, 2015; Shawn R Jeffery et al., 2006; Klein and Lehner, 2009b; Said and Masud, 2013; Sathe et al., 2013; Ukil et al., 2011; Zeng et al., 2011) arise from the afore-mentioned scenarios which endanger the quality of produced data. These problems affect the main components of the IoT system and relate to the following aspects:

- **Deployment Scale:** IoT is expected to be deployed on a global scale. This leads to an enormous heterogeneity in data sources as it will no longer come only from computers but rather from day-to-day's objects. Furthermore, the distributed aspect will be unprecedented. The huge number of devices accumulates the chance of error occurrence.
- **Resources constraints:** The things in the IoT (e.g. RFID tags) suffer generally from a severe lack of resources (e.g. power, storage, etc.). Their computational and storage capabilities do not allow complex operations support (e.g. cryptographic operations, etc.). Furthermore, they are usually battery-powered and they often operate with discharged batteries. Considering the scarce resources, data collection

Internet of Things

www.IoTiran.com

policies, where tradeoffs are generally made, are adopted which affect the quality and cleanliness of data.

- **Network:** Intermittent loss of connection in the IoT is rather frequent. In fact, IoT is seen as an IP network with more constraints and a higher ratio of packet loss. Things are only capable of transmitting small-sized messages due to their scarce resources.
- **Sensors:** Embedded sensors may lack precision or suffer from loss of calibration or even low accuracy especially when they are of low cost. Faulty sensors may also result in inconsistencies in data sensing. The casing or the measurement devices could be damaged due to extreme conditions like extreme heating or freezing which can also cause mechanical failures. The conversion operation between measured quantities is often imprecise (e.g. from voltage to humidity).
- **Environment:** The sensor devices will not be deployed only in tolerant and less aggressive environments. In fact, to monitor some phenomenon (e.g. weather), sensors are deployed in environments with extreme conditions (e.g. a mountain's summit). The maintenance of such sensors is rarely ensured considering the inaccessibility of terrains. In those conditions, sensors may become dysfunctional or instable due to many events (e.g. snow accumulation, dirt accumulation, etc.).
- **Vandalism:** Things are generally defenseless from outside physical threats. In addition, their deployment in the open nature makes them an easy prey for vandalism both from humans and animals. Such acts often result in rendering sensors dysfunctional which definitely affect the quality of produced data.
- **Fail-dirty:** It is a case where a sensor node fails, but keeps up reporting readings which are erroneous. It is a well-known problem for sensor networks and generally an important source of outlier readings.
- **Privacy preservation processing:** DQ could be intentionally reduced during the phase of privacy preservation processing.
- **Security vulnerability:** Sensor devices are vulnerable to security attacks. Their lack of resources makes it even harder to protect them from security threats (e.g. no support for cryptographic operations because of their high consumption of resources). For example, it is possible for a malicious entity to alter data in sensor nodes or RFID tags causing data integrity to fail.
- **Data stream processing:** Data gathered by smart things are sent in the form of streams to the back-end pervasive applications which make use of them. These data streams could be processed for a variety of purposes (e.g. extracting knowledge, decreasing the data stream volume to save up on the scarce resources, etc.). (Klein and Lehner, 2009b) argues that many data stream processing operators (e.g. selection) could, under certain conditions, affect the quality of the underlying data.

3.2 Layered distribution of factors threatening DQ

The aforementioned problems threatening the quality of produced data occur in different layers of the IoT system model. Various architectures were proposed for IoT (Atzori et al., 2012; Bauer et al., 2013; Kovatsch et al., 2012; Pujolle, 2006; Tanganelli et al., 2013; Vajda et al., 2011) that could be used as a referential to discuss the affected layers. It is worthwhile to mention that the resulting final positioning of these hazardous elements is greatly influenced by the chosen architecture. We consider the 3-layered architecture used in (Yan et al., 2014) composed of; (i) The Physical perception layer (PPL), (ii) the Network layer (NL) and (iii) the Application layer (AL).

The physical perception layer represents the sensing and actuation infrastructures. It is responsible for the generation of the huge amount of data used to represent the physical world in the cyber world. The Network layer groups heterogeneous network components responsible for processing and transmitting data. Finally, the Application layer provides ubiquitous services for users based on data feed from the Network layer.

While some of the aforementioned problems are limited to one layer, others have larger span over multiple layers (Fig. 2). The deployment scale spans across all layers of IoT architecture. For both PPL and NL, keeping infrastructures that meet the vision of IoT will be both challenging and source of problems. For AL, the huge deployment scale means more generated data (not necessarily from trusted or known third parties) that require more resources and efficient processing techniques. Moreover, pervasive applications in the AL need to process the received data stream to extract actionable knowledge mandatory in order to provide services. Further, the resources constraints are characteristics of the things which form the PPL. The intermittent loss of connection is a problem affecting PPL because of the scarce resources not allowing the things to maintain the data flux. It is also affecting the NL because of inefficiency of existing networking solutions in the context of IoT environment. Faulty and fail dirty sensors related problems concern the PPL considering that sensors are one of PPL's key building blocks. Environment-related problems affect the things-enabled PPL infrastructure and the networking infrastructure.

Security attacks could target components of any layer. Also, the privacy of gathered data from smart things needs to be ensured as it may be used to reveal the identity of their owner (e.g. a RFID tag that uniquely identifies a person and his whereabouts). The privacy preservation process may occur while collecting (i.e. within the PPL), transmitting (i.e. within the NL) or sharing data between ubiquitous applications and services (i.e. within the AL).

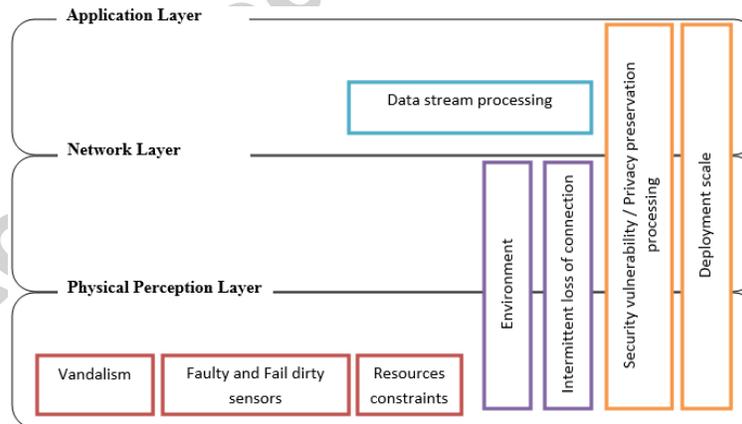


Fig. 2. Layered distribution of IoT factors threatening DQ

3.3 Impact on IoT DQ

In order to demonstrate how the previously mentioned IoT-related factors affect the quality of IoT data, we proceed by identifying patterns of how these factors impact various IoT DQ dimensions. In fact, DQ problems describe any difficulty affecting any DQ dimension, causing data to become entirely or partially unusable by not meeting user's requirements (Strong et al., 1997). Moreover, DQ problems manifest

not only as accuracy problems but surpass them to other DQ dimensions such as Completeness problems, Timeliness problems, etc (Wang and Strong, 1996). As a result, studying how DQ dimensions are affected in different scenarios will lead to creating a crisp image of how the overall DQ is affected. We qualitatively analyze how starting from a given context, where an IoT-related factor occurs (e.g. Resources constraints), certain DQ dimensions are affected. This creates a mapping between these IoT-related factors and DQ problems. In the following schemas depicted in this section, rounded rectangles represent the previously discussed factors associated with the IoT environment. We consider them as starting states. The dashed rectangles illustrate intermediate states that result directly from the aforementioned factors. Finally, the rectangles represent various DQ dimensions on which we want to determine the impact.

3.3.1 Impact of deployment scale, Failing-dirty, vandalism and environment on DQ dimensions in the IoT

Fig. 3 depicts the impact of the deployment scale factor on the quality dimensions accuracy and confidence. The vision of the IoT states that the smart things will ubiquitously be present around us. This unprecedented envisioned scale of deployment may have consequences on DQ. In fact, as we mentioned in the introduction, the number of connected objects is so large that it has already surpassed the number of persons on the planet (1). To achieve their main goal of harvesting data about their surroundings, these smart things use embedded sensors which are not ideal and therefore have a margin of error. For huge number of devices, the error margin may no longer be neglected and could eventually lead to an accumulated chance of error occurrence (2) which is synonym to lower accuracy in the gathered data. In order to reach this huge number of devices, different manufacturers should produce different types of smart things for different purposes (3). This means a highly heterogeneous landscape of devices. For an ubiquitous application which uses data sent by scattered smart things over which it has not necessarily some kind of control (i.e. third-party data streams), these received data should be used with precaution as it is uncertain whether or not they have been intact during collection, storage or transfer (4).

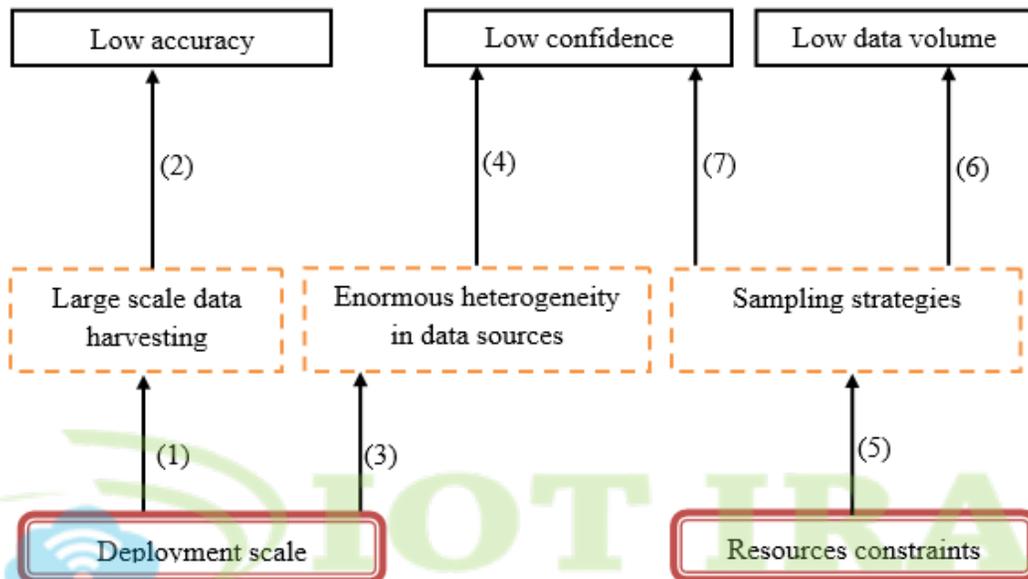


Fig. 3. Impact of deployment scale and resources constraints on DQ dimensions in the IoT

Fig. 4 depicts the impact of vandalism, environment and failing-dirty factors on DQ. There exists a plethora of applications using data about the real world (e.g. monitoring application). It might be unavoidable for some applications to deploy smart things in particularly harsh environments in order to gather data about some phenomenon of interest (1). Potential problems arise in this scenario. First, in aggressive environments (2), the deployed smart things are likely to get damaged for various reasons (e.g. dirt accumulation, snow accumulation, etc.) (3). Second, even in the event of a damaged smart thing, neither corrective nor frequent preventive maintenance might be available or provided (5) because of the inaccessible terrains (4) which generally characterize this type of remote deployment environments (e.g. mountain's summit). These unfortunate events might result in a dysfunctional smart thing. The same result could be caused by vandalism acts (7) which are likely to happen when taking in consideration that the smart things are generally both defenseless and deployed in open nature (6).

There are many direct consequences of dysfunctional smart thing on the quality of data it generates. In fact, a dysfunctional smart thing could fail dirty (8) and keep generating inaccurate data and feeding it to the back-end application (9). It could also become less available and could stop frequently (10) which will have a direct impact on how much data it generates about real world events (in comparison with how much data it is supposed to generate). Furthermore, not being able to deliver readings in time may negatively affect the timeliness of data (11). All these potential problems make it hard to put significant trust on data generated by a potentially dysfunctional smart thing (12).

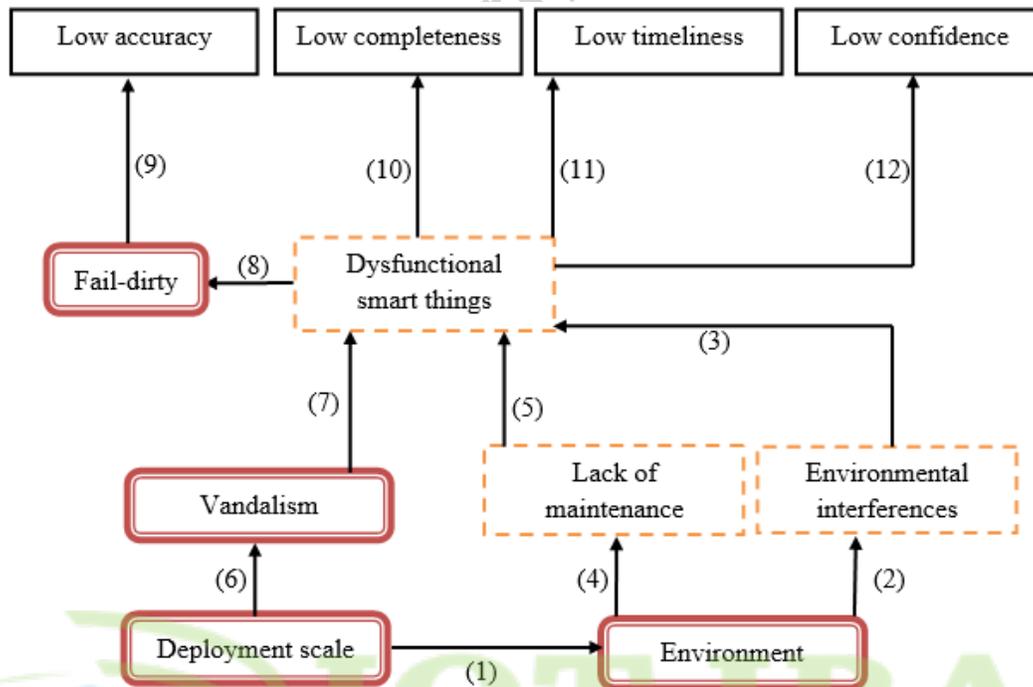


Fig. 4. Impact of deployment scale, vandalism, fail-dirty and environment on DQ dimensions in the IoT

3.3.2 Impact of resources constraints, unreliable sensors, network, security vulnerability and privacy preservation processing on DQ dimensions in the IoT

One major problem that confronts the IoT is that the smart things usually suffer from severe resources constraints which could affect the DQ as depicted in Fig. 3. Many approaches for saving up resources are adopted. Such approaches include sampling strategies used while gathering data considering that smart things cannot operate continuously (5) (i.e. required sleep mode to save energy (Jardak and Walewski, 2013; Uckelmann et al., 2011)). In many cases (e.g. monitoring natural phenomenon), the sampling means the discretization of continuous phenomena, i.e. only part of the available data is gathered (6). Also, statistical errors are introduced due to uncertainty about sampling estimates which lower the confidence about the received data (7).

Fig. 5 further depicts the impact of the resources constraints factor. In fact, the constrained resources prevent sufficient security protocols from being supported by smart things, as they usually require significant resource capabilities, resulting in a lack of a built-in security layer which makes smart things vulnerable (1). These security vulnerabilities could be used by unauthorized entities to gain access to the device (2). Once the device is hijacked, all forms of malicious commands could be executed ranging from altering stored readings (3), nullifying all or some attributes (4) to blocking readings transfer (5) which affects the accuracy, the completeness and the timeliness respectively.

The privacy (6) is a key security requirement to be ensured especially in the context of IoT where tiny devices could be used to breach one's privacy (e.g. a RFID tag that uniquely identifies a person could be used to remotely trace his movements). Privacy preservation processing techniques intend to blur or even break the link between sensitive data and the originate owner (i.e. the source) without critically affecting its capability to provide valuable insights about a certain phenomenon of interest (i.e. ensuring privacy while minimizing information loss). Various techniques for privacy preservation intentionally reduce DQ (7). Examples include aggregating data (i.e. reducing data volume) (8), adding noise (9) or falsifying (i.e. reducing the accuracy and confidence) geo-location data in order to hide the user's true location (Zheng and Zhou, 2011), reducing data accuracy (Aggarwal and Yu, 2008) (10), removing (sensible) attributes (11) and processing surveillance systems' video stream data in order to mask authorized persons (i.e. reducing the completeness and the accuracy of video stream data) (Wickramasuriya et al., 2004).

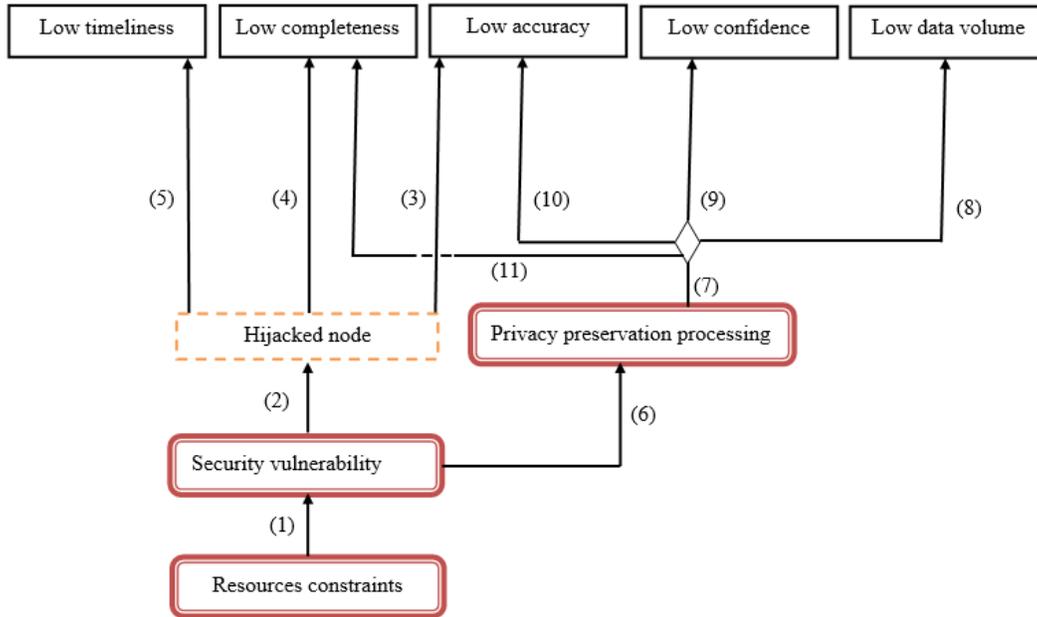


Fig. 5. Impact of Resources constraints, Security vulnerability and Privacy preservation processing on DQ dimensions in the IoT

Further, Fig. 6 shows other scenarios where resources constraints affect the DQ. In fact, most of smart things are neither capable of sending large messages (i.e. packets) nor capable of reporting frequently (1). As a result, only small-sized messages are being exchanged which could turn out to be insufficient to report all available data (2). Moreover, because of these scarce resources, smart things will frequently go through sleep mode in order to conserve energy (3). However, the IP protocols, forming the backbone of IoT connectivity, are not adapted to these sleep modes and require the smart things to be always operational (4) (Aggarwal et al., 2013). This incompatibility results in an instable connectivity and intermittent loss of connection which translates to a high ratio of packet loss (5). The completeness of data could be highly affected by those lost packets (6) and even if some kind of packets reception acknowledgement and recovery mechanisms are in place, it might be too late to resend the lost packet as it might have already become outdated (7).

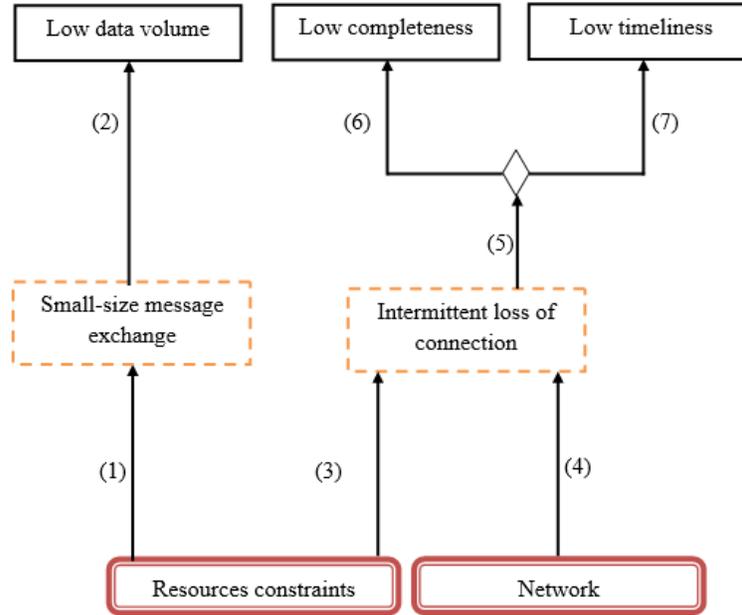


Fig. 6. Impact of resources constraints and network on DQ dimensions in the IoT

Fig. 7 shows the impact that sensors could have on DQ. In fact, considering the number of deployed objects envisioned by the IoT, it would certainly be less expensive to use cheap devices over expensive ones. However, there are many problems associated with cheap devices mainly because their components tend to be less reliable (1). As an example, low cost embedded sensors are more prone to loss of calibration and lack of precision which results in the measured values being less accurate and more uncertain (2). Another factor that may introduce errors in the reported measurements is the conversion process, often imprecise (4), used to convert between measured quantities (3).

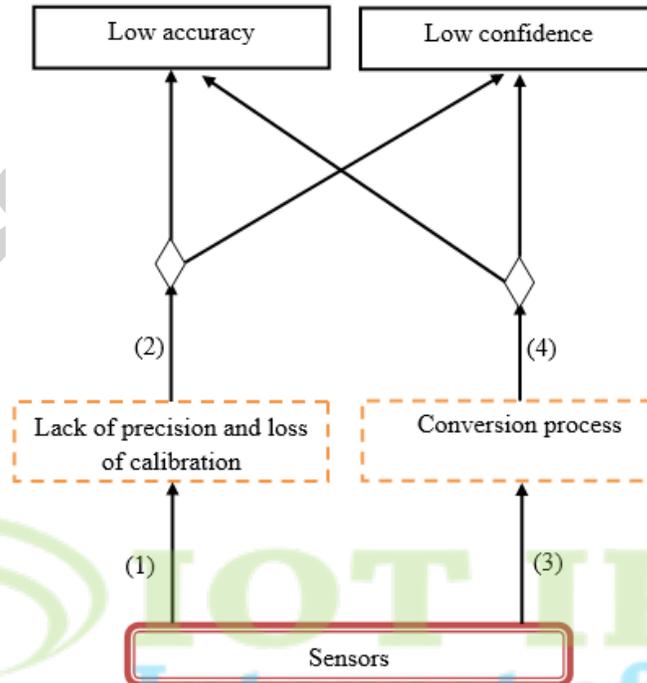


Fig. 7. Impact of unreliable embedded sensors on DQ dimensions in the IoT

3.3.3 Impact of data stream processing on DQ dimensions in the IoT

As we mentioned earlier, (Klein and Lehner, 2009b) studied how different data stream processing operators could affect the quality of the data they process. As it turned out, there exists an influence of the data stream processing on the quality of the underlying data. We summarize some of the paper's discussed cases in Fig. 8. After identifying four classes of data stream processing operators, namely Data-modifying operators (1), Data-reducing operators (2), Data-generating operators (3) and Data-merging operators (4), the authors studied the impact of different operators (e.g. Selection, Unary algebraic operators, etc.) belonging to each defined class on DQ. In fact, some operators (e.g. Timestamp-Join of Asynchronous streams) include sampling operations which introduce statistical errors (5).

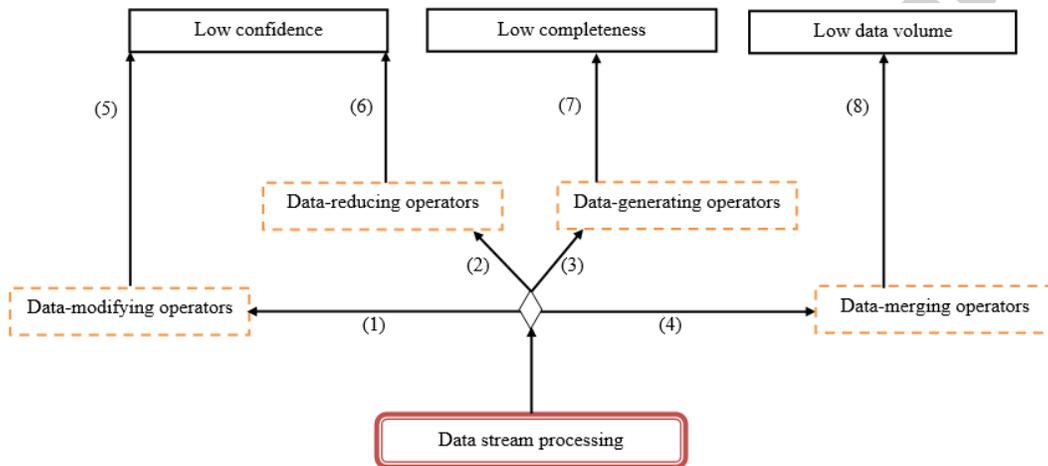


Fig. 8. Impact of data stream processing on DQ dimensions in the IoT

Other operators such as the Selection operator could also introduce statistical errors (e.g. when an item that shouldn't be selected, does get selected) (6). Moreover, inserting data items (e.g. inferred from existing one using interpolation) into data stream lowers the completeness (7). Finally, aggregating data items reduces data volume (8).

3.3.4 Impact of different IoT problems on Ease of Access, Access security and Interpretability DQ dimensions

Both Fig. 9 and Fig. 10 show the impact of the previously mentioned IoT-related factors on Ease of Access, Access Security and Interpretability dimensions. In these figures, the dotted lines indicate that we omitted some intermediate states which have already been shown in Fig. 3, Fig. 4, Fig. 5, Fig. 6, Fig. 7 and Fig. 8.

As depicted in Fig. 9, dysfunctional smart things may no longer be available to answer data retrieval queries rendering the stored data unavailable for the requesting applications (1). This unavailability could also be caused by the intermittent loss of connection as a stable connection for retrieving data may be hard to maintain (2) or that the supported size of exchanged message is not sufficiently enough to deliver all the available data within a reasonable time interval (3).

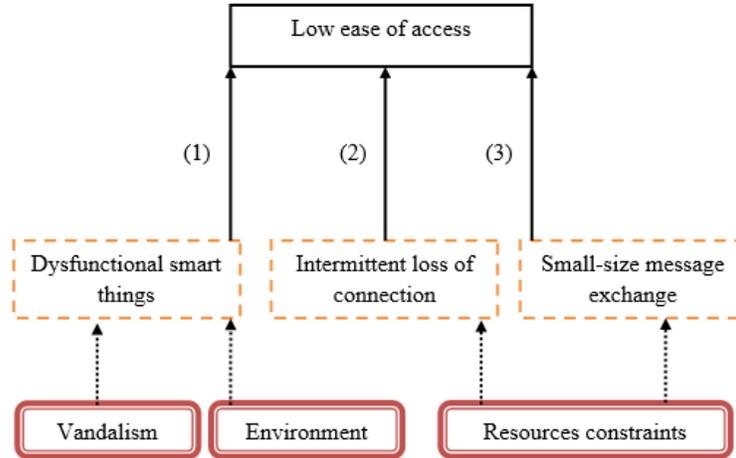


Fig. 9. Impact of different IoT-related factors on Ease of Access DQ dimension

Further, Fig. 10 reports other impact scenarios. In fact, the huge amount of available data could significantly slow down the retrieval process performances. Also, too much data could become a barrier to retrieving specific data of interest. Both these aspects affect the Ease of Access dimension (1). Moreover, when a smart thing is hijacked, the access control rules could be maliciously changed as to deny authorized access, thus both the smart thing and the data it has gathered become unavailable for authorized entities (2).

The security vulnerabilities of the smart things caused by their lack of resources also affect the Security Access dimension as it becomes more difficult to protect the privacy and confidentiality of data. More specifically, when a smart thing is hijacked, both the privacy and the confidentiality of its stored data could be breached as the attacking malicious entity may have already gained full control of the device (3).

One of the key features of the IoT is that large numbers of devices will be able to harvest data and provide them to ubiquitous applications. However, if each provided data stream has a different structure (4) and the ubiquitous applications have no previous knowledge about it (which is likely as data streams could be produced by third parties), merging them would not be intuitive or out of the box. In such case, the received data would be hardly interpretable and would serve no real purpose.

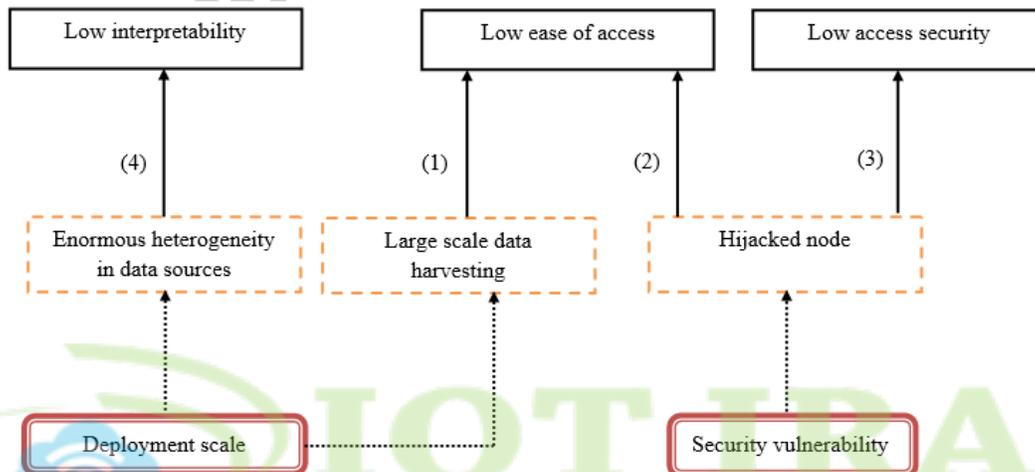


Fig. 10. Impact of different IoT-related factors on Ease of Access, Access security and Interpretability DQ dimensions

4. MANIFESTATION OF DQ PROBLEMS IN IOT AND THEIR SYMPTOMS

Data in the IoT are vulnerable to many risks affecting their Quality. Data suffering from quality problems fail to represent the reality and could have negative effects both on the decisional and operational levels of any business or organization (i.e. data consumer) (Batini and Scannapieco, 2006). In this section, we present in what form do DQ problems manifest. We also discuss their symptoms by associating each manifestation class with the set of its characterizing affected DQ dimensions.

4.1 DQ problems' manifestation classes

In (Jeffery et al., 2006a), two manifestation classes have been presented for “dirty-data” (i.e. low quality data):

- **Dropped readings:** The delivery of things' readings is usually inferior to the requested readings required by pervasive applications. In fact, the ratio of successful delivery is typically low due to scarce resources and intermittent communication that cause a drop in the reporting efficiency. The ratio of dropping could be measured with metrics such as the epoch yield (Jeffery et al., 2006a) defined as the total reported readings as a fraction of total requested readings by an application.
- **Unreliable readings:** Impreciseness, calibration failure and fail-dirty nodes among other reasons cause things' data to be unreliable.

Other IoT DQ problems were reported in the literature and they include:

- **Multi-source data inconsistencies** (Ma, 2011; Mishra et al., 2015; Rao, 2016; Vongsingthong and Smanchat, 2015; Wang et al., 2012): Data in IoT come from a number of different objects (e.g. RFID and sensors) and in a variety of structured, semi-structured and unstructured formats (e.g. text, numerical, video, etc.). This makes non-uniformity and inconsistency big challenges when handling IoT data. In fact, most ubiquitous services enabled by the IoT use data from different sources which require extensive integration and fusion. Moreover, the dynamic nature of IoT itself could cause inconsistencies in the generated data in cases such as addition or removal of smart objects from existing deployments (Vongsingthong and Smanchat, 2015).
- **Data duplication** (Amadeo et al., 2014; Li et al., 2013; Mishra et al., 2015; Rong et al., 2014): When a data consumer (e.g. a pervasive application) sends data retrieval requests, many data producers (e.g. sensor nodes) may report similar harvested data. In other cases, a data producer could report similar data as a response to various data retrieval requests (e.g. similar video frames recorded from an environment that did not change over time, RFID tags readings). Considering the scope of IoT, the huge number of smart objects already/to be deployed and their constrained resources, these duplicated data could introduce significant costs for transmission, handling and storing.
- **Data leakage** (Singh et al., 2015; Weinberg, 2004): Data leakage occurs when an application retrieves or store more data than needed. Considering the ubiquity of IoT, this issue could be very threatening to user's privacy because data in IoT carry

more insights about our daily life and routines than, probably, any other medium ever had.

- Multi-source data time alignment (Shah, 2016): Most ubiquitous services require integrating data from multiple sources (both dynamic and static) to provide appropriate services. However, issues related to the time-alignment of these data sources, to extract significant insights, usually rise due to the spectacular rate of generating real-time data in IoT and the huge number of available data producers.

Finally, it is worth noting that DQ problems existing in semi-structured and unstructured data, which are extensively used in IoT, still need more investigation (Wang, 2016) and may reveal other classes of DQ problems.

4.2 Symptoms of DQ dimensions difficulties associated with DQ problems classes

In the previous chapter, we used DQ dimensions as a metric to study the impact on DQ. We also noted that any DQ problem could be expressed as difficulties at the level of DQ dimensions. In this paragraph, we specify the symptoms (i.e. affected DQ dimensions) of each of the DQ problems classes defined above. The resulting mapping is depicted in Fig. 11.

Unreliable data as a DQ problem class represent the uncertainty inherent to data items due to various factors. This uncertainty is related to the extent the measured data items' values represent the true values with respect to the measurement's accuracy and preciseness. Both the accuracy and confidence dimensions could be used to profile these data items. Items suffering from this DQ problem will be characterized with poor accuracy and confidence.

Both low completeness and low data volume are key symptoms of the Dropped Readings DQ problem class as they both translate to the ratio of dropped/missing values (e.g. NULL values) in the reported data stream.

Timeliness deficiencies represent a special DQ dimension as they could be seen as a key symptom of both the dropped readings and the unreliable readings DQ problem classes. In fact, on one hand, an outdated reading (i.e. not arriving in time with respect to the use requirements) essentially means that one requested reading by the application could not be delivered on time. As a result, even when this reading arrives, it will be of no use and will be ignored/dropped either explicitly or implicitly (e.g. by not incorporating them into the data processing). On the other hand, an obsolete reading in a dataset could affect the soundness of the decision making. Let us take, for example, a decision making process that takes as an input the readings from the last thirty seconds and derives actionable decisions (e.g. adjust temperature in a factory production line). If an obsolete reading (in our case, a reading measured more than thirty seconds ago) has finally arrived and is used in the input dataset, the derived decision is no longer valid as it depends not only on the state of the last thirty seconds (as required by the user or the application-defined rules) but also on readings from an interval of non interest for the user/application. As a result, obsolete readings could be seen as outliers (with respect to the application's timeframe of interest) that are unreliable and should not be used for later data processing (e.g. business decision making).

Multi-source data-related problems manifest generally in the form of low consistency. Moreover, data generating objects use various data formats causing significant data presentation problems resulting in a low interpretability and low interoperability between the incoming data streams. As regards data duplication, it mainly affects the data volume by making it unnecessary and costly huge due to redundant readings. Further, data leakage shows a low level of access security to generated data in smart objects allowing certain applications to pull more data than

they should be authorized to. Finally, the mutli-source data time alignment is essentially a time related problem and manifests in the form of low timeliness.

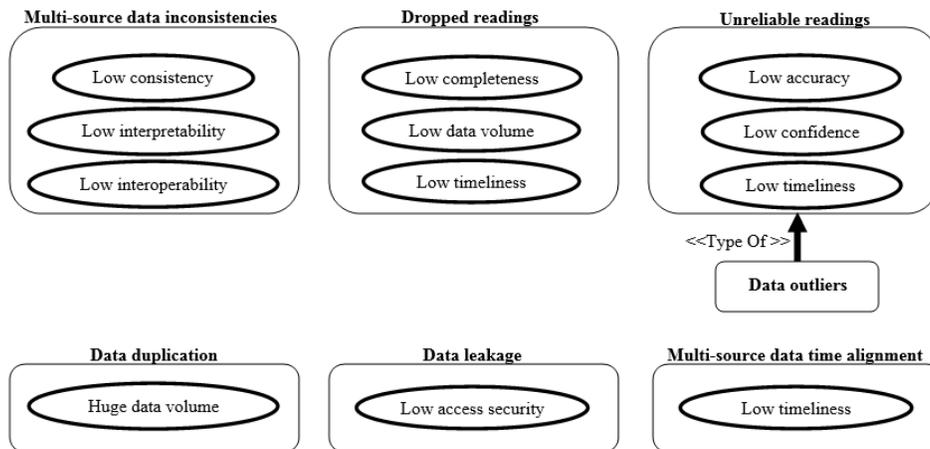


Fig. 11. DQ problems' symptoms

5. DATA OUTLIERS

Data produced in the context of IoT by the things are generally unreliable. Data outliers are one of the major manifestations of data uncertainty. In this section, we study data outliers as a specific representation of the “Unreliable Readings” DQ problem class. We start by defining the concept of a data outlier. We then discuss their types. Finally, we analyze their impact in the IoT context.

5.1 Definition of data outliers

Outliers or anomalies belong to the class of unreliable readings as depicted in Fig. 11. They are readings that are outside what is considered as a “normal state” (represented by a model for example) (Javed and Wolf, 2012). In (Branch et al., 2009), outliers are considered “events with extremely small probabilities of occurrence”. They are also seen as “points in a data set that are highly unlikely to occur given a model of the data.” (Otey et al., 2006). Following the same logic, (Chandola et al., 2009) defines anomalies as “patterns in data that do not conform to a well-defined notion of normal behavior”. There exist other formal metric-based definitions such as distance-based outliers (DB-outlier) (Knox and Ng, 1998), for which “an object O in a dataset T is a DB (p , D)-outlier if at least fraction p of the objects in T lies greater than distance D from O ”. Fig. 12 shows an example of outliers in a dataset. The values within the regions $R1$ and $R2$ are considered normal, while the values outside these 2 regions are considered outliers.

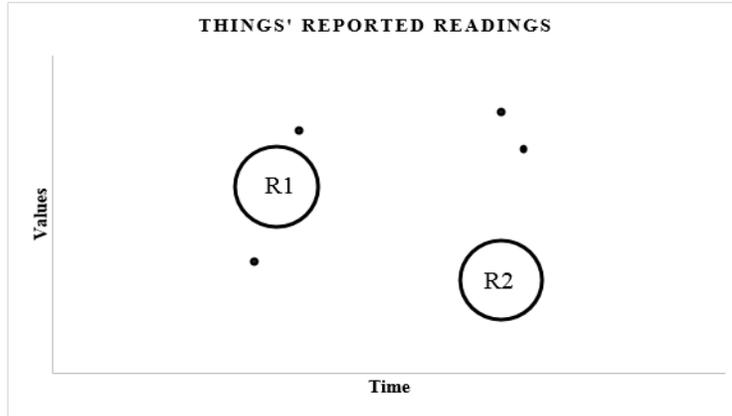


Fig. 12. Example of outliers in a reading dataset

5.2 Types of outliers

Outliers are elements that significantly differ from other elements in a dataset. This does not automatically mean that they represent errors. In fact, outliers may translate to important information.

Based on the carried underlying knowledge, an outlier could represent (Zhang et al., 2010):

- An error: A value generated due to a system dysfunction (e.g. a node failure).
- An event: A value generated because of a sudden or extreme change in the monitored phenomena (e.g. a passing hurricane). It represents an extreme (legitimate) reading.

Another classification of outliers is presented in (Chandola et al., 2009):

- Point anomaly: Represents a single value that differs greatly from other values in a dataset.
- Contextual anomaly: Represents a value that, depending on the context, could be considered an outlier or not. The same value could be an outlier in one context but not in another.
- Collective anomaly: Represents a collection of related values which differ largely from the rest of values in the dataset. The occurrence of the whole cluster of values is what is considered outlier and not necessarily the individual values.

5.3 Impact of outliers in the IoT

In the IoT, data gathered by things (e.g. body sensor) will serve as an input for data mining in order to extract insights about a given monitored phenomena (e.g. environment, home, health, etc.). Upon these insights, decision will be made in a pervasive manner (e.g. call emergency, fire alert, etc.). It is clear that insights extracted from “dirty data” (i.e. unreliable data) are probably erroneous and thus the decisions to be made are likely unsound (e.g. high rate of false positive and false negative). As an example, in the RFID-enabled library scenario (Jeffery et al., 2006a), almost half of the emitted alerts are false positives due to unreliability of raw data.

The importance of accuracy and reliability of data is even higher when exploited in applications which involve or affect human lives. Examples are given in (Burdakis and Deligiannakis, 2012) which include a scenario where phenomena such as Tsunamis and avalanches are monitored in order to quickly evacuate endangered zones. Another scenario involves monitoring fire in forests in order to quickly react and take appropriate procedures. In addition to applications in component monitoring where, in order to protect expensive systems and avoid damages, the data

about the state of components should be accurately reported. Any failure to deliver accurate data may compromise whole systems or even put peoples' lives at risk.

While decisions made based on erroneous raw data are likely unsound, this is not the case, if the outliers report rare events in which case they represent a mine of gold for the pervasive applications: "One person's noise is another person's signal." (Knox and Ng, 1998). In fact, for some applications, outliers (representing events) are far more important, with respect to the knowledge discovery standpoint, than the "common behavior" as they represent rare events (Knox and Ng, 1998) (e.g. fire in a forest).

Data trustworthiness is crucial for user engagement and acceptance of IoT services, thus, for a successful large scale deployment of the IoT paradigm. This is shown in (Yan et al., 2014) where data collection trust and accuracy represent the major concern of the Data Perception Trust as part of a holistic trust management approach for the IoT.

Simulations or reproducible experimentations are an effective way to better understand IoT systems and its challenges. In fact, many Testbeds for IoT experimentation, such as FIT-Equipex (Papadopoulos et al., 2013), already exist and are operational. A number of other existing (public and private) Testbeds are surveyed in (Gluhak et al., 2011). However, to further investigate the impact of data outliers, we report two real-world case studies of DQ issues' impact on the e-health application domain.

E-health refers to internet (and other related technologies) –enabled services which aim to provide and improve healthcare (Eysenbach, 2001). Data play, within Medical Information Systems, a fundamental role as the source of information and knowledge. However, as we have previously stated, decisions taken based on poor quality data are probably erroneous and this kind of uncertain decisions cannot be allowed or tolerated in the e-health domain considering the patients' life involved.

The first case study (Rodríguez et al., 2010) investigates the impact of DQ issues on e-health monitoring applications. In this work, and in order to identify DQ problems affecting DQ criteria (e.g. accuracy, precision, currency, accessibility and consistency) that are argued to be crucial to provide appropriate assistance for the patient, three levels of data management are defined with respect to a scenario of cardiac monitoring application: Data collection, data processing and data discovery. For example, in data collection level, problems are mainly related to body sensors performance, the volume of data to pre-process and the quality of communications.

The second reported case study (McNaull et al., 2012) investigates poor DQ impact on Ambient Assisted Living systems (AAL) (resulting from the convergence of Ambient Intelligence and Assisted Living technologies). AAL provides support (e.g. monitoring health and wellbeing) for people in their homes. In this work, poor DQ is argued to alter the presentation of occurring events, thus, preventing these systems from providing adequate support for users and causing erroneous reporting of a patient health, inefficient management of in-home environmental conditions, etc.

E-health applications are among the most critical IoT applications considering the human life factor involved and, as a consequence, do not tolerate uncertainty in DQ.

5.4 Context-awareness and data outliers

Context-awareness is fundamental in IoT applications and a key feature in many industrial IoT products (Perera et al., 2015). The concept of "context" has been defined in (Abowd and Mynatt, 2000; Dey and Abowd, 1999; Pascoe, 1998; Schilit et al., 1994). It refers to information used to provide adequate services, i.e. the expected

services under certain circumstances, to a user based on the representation they construct about the current situation and all the entities involved (Gallacher et al., 2013). Raw data (e.g. sensor data) are the base to extracting context. For example, the data retrieved from temperature sensors are raw data. The knowledge inferred from these data about whether the weather is hot or cool, is the context.

Four phases compose the context life cycle: Context acquisition, Context modeling, Context reasoning and Context dissemination (Perera et al., 2014). Context-related actions (e.g. context acquisition, etc.) could be performed within the application or outside (e.g. in a dedicated infrastructure such as middlewares like (Guo et al., 2010; Wang et al., 2009)) (Hu et al., 2008). The data cleaning should occur, at the latest, in the Context acquisition phase as to ensure that the inputs of the upcoming phases are sound.

It is clear that the more accurate raw data are, the more relevant and accurate the obtained context is. In fact, the quality of context (QoC) depends on the quality of physical sensors and initial data (Bellavista et al., 2012). In (Dey and Abowd, 1999), five pillars for building context are specified: Who, What, Where, When and Why. If one of these axes is not present or poorly built, then the whole context will collapse. Now, let us consider that our raw data or a portion of it is of poor quality. By further processing raw data, we could retrieve the Who, the What and the When components. However, the “Where” component retrieved does not accurately describe the location because of the deficiency of initial data. As a result, the reason, i.e. the Why, a situation is occurring cannot be determined and the extracted context will probably trigger unsound actions which defies a basic principle of the pervasive aspect of IoT which is providing the right services in the right time for the right person.

As an example, a device equipped with multiple sensors allows parents to monitor the state of their child and his location. If the context built based on data received from that device is wrong because some sensors failed dirty and are reporting inaccurate readings, then the parents will have a wrong idea about the situation of their child. In fact, the child may have already left the safe zone that the parents have determined, but because of the device that kept reporting erroneous positional data, neither the parents nor the monitoring application have triggered the alarm of the child been missing as the context they had did not suggest so.

6. DQ ENHANCEMENT APPROACHES

In contrast with the conventional Internet, IoT data come from things rather than persons, thus, the programming’s golden rule “Never trust user input”¹, should evolve to “Never trust things input”. This is justified by the inherently afore-mentioned uncertainty and inconsistency of sensor data. To avoid costly consequences of low DQ, techniques to pre-process data and improve their quality are needed. In this section, we present five major DQ enhancement techniques namely outlier detection, interpolation, data integration, data deduplication and data cleaning. We present their main principles, their general processes and the DQ aspects they enhance. Moreover, we propose an extended taxonomy of criteria which we use to compare different data cleaning techniques and outline their characteristics and their fitness to be used in the context of IoT.

6.1 Outlier detection

6.1.1 Outlier detection's process description

During the process of outlier detection, elements that differ from what is considered normal are discovered. The end-goal is to either suppress or highlight outliers (Branch et al., 2009). On the one hand, once outliers (i.e. elements believed to be unreliable or extremely suspicious) are discovered, further operations could be easily taken such as suppressing discovered outliers (i.e. eliminating unreliable elements). On the other hand, highlighting outliers is used while looking for rare events and patterns underlying in a dataset in specific domain (e.g. fraud analysis).

Outlier detection is closely related, however differs from noise removal and noise accommodation. It is also closely related to novelty detection where novel patterns in dataset are mined and incorporated into normal model (Chandola et al., 2009).

6.1.2 DQ enhanced aspects

Outlier detection helps improve the overall quality of datasets by making them more consistent. Moreover, outlier detection represents the first phase for handling instances of the Unreliable Reading DQ problem class. In fact, data cleansing processing adopts the approach of suppressing discovered outliers in order to increase DQ. Consequently, the accuracy and reliability of data processing results are also increased (i.e. more sound decision-making). It is worth noting that the accuracy at the level of single data items itself is not increased as it is only related to data generating sources and cannot be improved by data processing operations (Klein and Lehner, 2009a).

6.1.3 DQ dimensions in outlier detection

Metrics used in outlier detection techniques focus on highlighting the difference between data values in order to identify outliers (e.g. a value not fitting an established model (Javed and Wolf, 2012)). To the best of our knowledge, none of the techniques use metrics that highlight the intrinsic DQ in processed datasets i.e., to use DQ dimensions or attributes assessment to identify data values that represent outliers. In (Klein et al., 2007), a framework based on Data Stream Management Systems (DSMS) and Relational Database Management Systems (RDBMS) metamodels extensions has been proposed to ensure an end-to-end management of DQ from capturing to persisting. Data values, within each window, are associated with their DQ dimensions values. In theory, these DQ values could be used as criteria to classify data. Data values could be considered outliers or of poor quality when their corresponding DQ values do not validate some specified threshold. However, these same DQ dimensions values, used for evaluating data, may themselves be unreliable or become insignificant under certain circumstances. An example would be the Accuracy dimension used in the aforementioned framework which is retrieved from the manufacturer-specified sensor's measurement precision class. If the sensor fails dirty under any circumstances, then the Accuracy dimension is rendered insignificant and unreliable.

6.2 Interpolation

6.2.1 Definition

Interpolation consists of inferring missing values based on other (available) values. In the context of data stream, it represents an estimation of missing data stream

attributes or tuples (due to sensor malfunctions or loss of connection, etc.). For example, if a tuple j is not received and it is required for later data processing, its value could be estimated knowing the values of tuples $j-1$ and $j+1$ (We suppose that both tuples $j-1$ and $j+1$ were correctly received). Many methods for interpolating data points exist including linear interpolation, polynomial interpolation, etc. Interpolation is used in a variety of domains such as (Hofstra et al., 2008; Štěpánek et al., 2011).

6.2.2 DQ enhanced aspects

Missing values represent gaps in available data about a certain entity or phenomena of interest for the user. As knowledge deriving processes use these datasets as input, these gaps could also lead to incomplete knowledge or wrong decisions which means that missing values could lead to a decrease in DQ. Further, a user-defined rule could specify a threshold on the size of data input (i.e. data volume) or a constraint on the number of Null values (i.e. completeness) for data processing which should be fulfilled in order to carry on with later phases of data processing pipeline. As it is not rare to not receive a requested value, interpolation methods could be used to meet user requirements (e.g. providing timely (predicted) data items in lieu of lost ones) and to overcome instances of the Dropped readings DQ problem class by inferring the missing values. In fact, interpolation is a data generating approach and as such it does improve the data volume DQ dimension (i.e. increases the available data items). However, interpolation has an opposite effect on the completeness DQ dimension which by definition is the ratio of non-interpolated items to all available (i.e. both non-interpolated and interpolated) data in the considered stream window. This situation could be described as an optimization problem that should be resolved to find the optimal compromises between these two DQ dimensions with the constraint of fulfilling the user-defined DQ requirements (Klein and Lehner, 2009a). Moreover, an important factor to take into account when choosing an interpolation technique is the accuracy of interpolated values (Chaplot et al., 2006) which should also fulfill the user-requirements.

6.3 Data integration

6.3.1 Definition

IoT data come from a heterogeneous landscape of smart objects. In order to be used, these data need to overcome their structure differences and inconsistencies to become truly beneficial for the ubiquitous services. Semantic-based integration approaches falls under the semantic-vision of IoT and they aim to make the integration and interoperability of IoT sensor data more achievable. (Aggarwal et al., 2013) reports two such approaches namely the Open Geospatial Consortium's Sensor Web Enablement initiative and the World Wide Web Consortium (W3C)'s Semantic Sensor Networks Incubator Group (SSN-XL) initiative. Both these initiative proposes a suite of components and services specifying standardized mechanism to ensure ease of understanding and efficient interoperability of sensor data. Such components include ontologies, annotations, metadata, web service interfaces, etc. Moreover, frameworks such as the Resource Description Framework (RDF) (Staab and Studer, 2007) and the Web Ontology Language (OWL) (Bechhofer, 2009) are providing standardized mechanism to describe data in order to make tasks such as searching, retrieval and processing more straight forward. Further, the Linked Data is a very promising approach to ease data integration and retrieval in IoT (Qin et al., 2015). As an example, a semantic data integration framework using Linked Data principles and semantic web technologies was proposed in (Nagib and Hamza, 2016). Other

approaches consist of building middlewares to abstract the underlying physical sensing layer and ease data integration such as the OpenIoT project (Soldatos et al., 2015) and the Global Sensor Networks (GSN) project (Aberer et al., 2006).

Other solutions for data integration designed for specific IoT-domain application were also proposed. In fact, (Petrolo et al., 2016) designed an architecture for integrating smart objects in the context of a Smart City scenario. In addition, a Service Oriented Architecture (SOA) paradigm was also proposed to abstract heterogeneity of smart things and enhance their interoperability in the context of e-health applications (Włodarczyk et al., 2016).

6.3.2 DQ enhanced aspects

Data integration solutions mainly focus on resolving the presentation inconsistencies between the various data streams. Also, they provide means to improve the interpretability of data (e.g. by using annotations) and their interoperability by abstracting heterogeneous objects specifications and presenting pervasive application with unified interfaces to search, retrieve and process data.

6.4 Data deduplication

6.4.1 Definition

Data deduplication is a data compression mechanism aiming to reduce data handling's resources consumption by reducing the amount of available data through removing of duplicate data items and replacing them with a pointer to the unique remaining copy. A comparison of existing deduplication techniques is presented in (Mandagere et al., 2008). Moreover, (Sethi and Kumar, 2014) proposed a methodology to leverage the Hadoop Framework with deduplication capabilities. In a more IoT-specific context, (Li et al., 2015) proposes a technique for video deduplication with privacy preservation considerations. Finally, (Yan et al., 2016) proposes deduplication techniques for cloud stored encrypted data.

6.4.2 DQ enhanced aspects

Data deduplication is quite simply a removal process of redundant data items. As such, it mainly reduces the amount of data and affect the data volume DQ dimension.

6.5 Data cleaning

6.5.1 Data cleaning benefits and phases' pipeline

Data cleaning is part of the data's life cycle as described in (Sathe et al., 2013) alongside data acquisition, query processing and data compression. It represents an important task in data processing. Data cleansing is not a new process specific to the IoT context. It has already been defined as a process for database systems in (Maletic and Marcus, 2000) where it is composed of 3 main phases: (i) Determination of error types, (ii) Identification of potential errors and (iii) the correction of identified (potential) errors. It is also very common for managing enterprise data in the context of data warehousing (Jeffery et al., 2006a). Data cleaning is widely studied under the "things-oriented" vision of the IoT (Aggarwal et al., 2013). While outlier detection is limited to discovering outliers, data cleaning goes a step further and suppresses the discovered elements. In general, and in order to handle different types of problems (unreliable and dropped readings DQ problem classes instances), data cleaning techniques have a larger scope than both outlier detection and interpolation. In fact,

data cleaning techniques usually incorporate built-in capabilities for interpolation and outlier detection, thus, affecting all the DQ dimensions affected by both sub-components (Fig. 13). It is worth noting that while the majority of affected DQ dimensions are enhanced, others could be affected negatively (e.g. the interpolation decreasing the completeness of data as shown in Fig. 13) because of existing trade-offs between certain DQ dimensions.

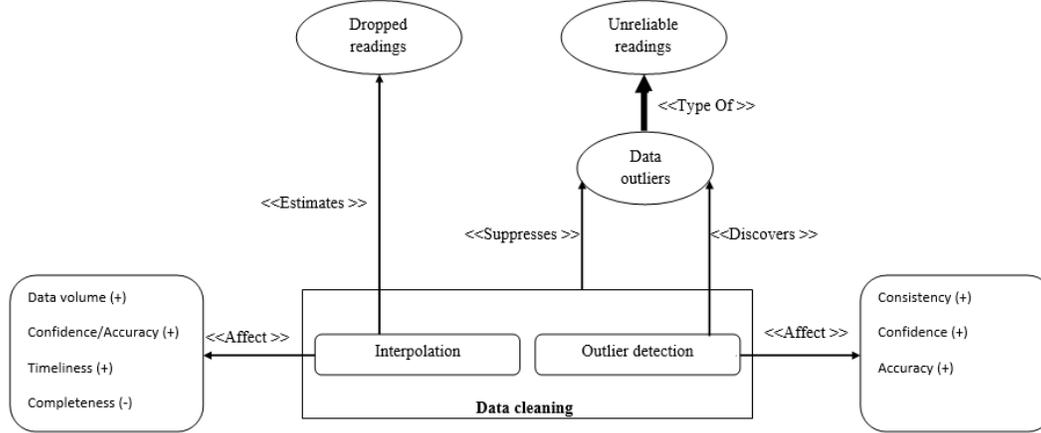


Fig. 13. Built-in capabilities of data cleaning techniques

Further, although existing applications do typically implement inner mechanisms for data cleaning within their logic considering the unreliability of data, their support for data cleaning is limited, application-specific and imposes post-processing overheads which increase development and deployment costs (Jeffery et al., 2006a). Providing a data cleaning system would help applications to concentrate on their core logic without worrying about data reliability post-processing overheads.

6.5.2 Data cleaning system's general architecture

In a data cleaning system architecture (Fig. 14), four main components are generally present as described in (Sathe et al., 2013):

- The user interface: Responsible for interactions (inputs and outputs) between the system and the end user.
- The stream processing engine: Maintains the flow of data incoming from physical world and operates as a data cleaning platform.
- The anomaly detector: Searches for outliers in datasets.
- The data storage: Stores raw data and cleaned data. Both kinds of data are kept because the raw data could be used in future data cleaning processes.

The data cleaning process could be implemented as part of the middleware layer (Aggarwal et al., 2013) that hides the complexity and details of the physical perception layer from the application layer. The major role of a data cleaning process, as described in (Branch et al., 2009), is to identify and suppress outliers in order to increase DQ. The identification task is carried out by the outlier detector component which could be argued to be the core component of the data cleaning system. Furthermore, from a design standpoint, multiple outlier detection techniques could be used in parallel as it was demonstrated in other domains such as in network traffic anomaly detection where a parallel design outperforms all the individual algorithms (Shanbhag and Wolf, 2008).

Datasets may belong to a variety of domains (e.g. meteorology). The underlying knowledge presented by these datasets is specific to their respective domains. This is a major concern because in order for a data cleaning process to purify datasets, a domain-specific knowledge is often argued to be necessary for optimal results (Maletic and Marcus, 2000). In the afore-mentioned data cleaning architecture, the user interface plays the role of an interface for capturing domain-specific knowledge (e.g. confidence, thresholds, etc.) from the end users whom also may be required to decide whether a detected outlier is a correct value or an actual error.

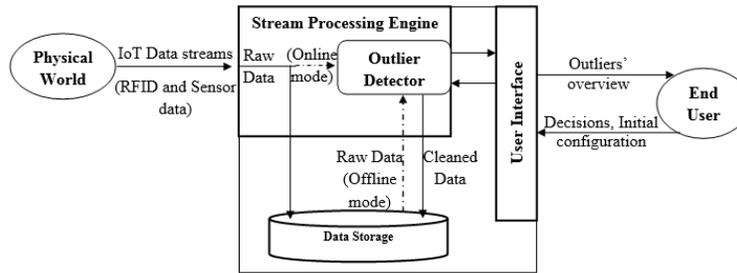


Fig. 14. A Data cleaning system's architecture (Adapted from (Sathe et al., 2013))

6.5.3 Comparison's Taxonomy

There are many data cleaning and outlier detection techniques. In order to better outline the characteristics of each approach, we adopt the following criteria on which we found our comparison (summarized in Table 5 and Table 6):

- **Approach type:** Different techniques use different approaches to perform data cleaning tasks. For example, the model-based approaches use well-established mathematical models to represent the datasets, while the declarative-based approaches use a higher-level abstraction queries to describe the cleaning process. To give more insights about how a mathematical model is built and used for assessing DQ, the schema in Fig. 15 summaries the case of the model-based technique in (Javed and Wolf, 2012) which we discuss in more details in 6.5.4.1. First of all, the initial data are broken up to Training Samples and Testing Samples. The former are used as input to create the model (e.g. using multiple regression) whereas the latter are used to avoid over-fitting problems. A “good” model performs well on both Training and Testing samples. Once created, real-life data streams are tested against the model and the result is given as a binary output; a “normal data item” or an outlier (with respect to the used model).
- **Cleaning Scope:** Some techniques offer a holistic framework for data cleaning while others focus on the outlier detection component.
- **Data stream Sources:** RFID and sensor technologies are the main enablers of the IoT vision. They are the major sources of data streams in the context of IoT. The surveyed techniques are classified based on their capabilities to process RFID- and sensor-enabled data.
- **Data characteristics:** As it was mentioned in 2.2, data in the IoT exhibit many properties. Many of the surveyed techniques make use of these characteristics to perform their tasks.
- **Operating mode:** There are 2 operating modes: (i) online mode and (ii) offline mode. In the first mode, data stream is processed as soon as it is captured in a real-time fashion. No trace files storage or processing is required which results in a simpler

system design. The latter mode processes data after their arrival and storage. In this case, trace files, typically voluminous due the amount of data generated, are stored and processed periodically or on demand.

- **Event/Error Separation:** Outliers could represent either errors or important events. The ability to distinguish the underlying knowledge of an outlier could help reduce the intervention of the human factor.
- **Automatic:** The degree of human intervention in the data cleaning process varies. Some techniques need minimum human intervention (e.g. automatically discover an underlying model in datasets), others require more human intervention (e.g. specifying threshold and query for every step of the cleaning pipeline).
- **Domain-agnostic:** Depending on the need for prior (domain-specific) knowledge, techniques are either domain- agnostic, i.e. usable for various domains, or domain-specific, i.e. they are designed for the use in a specific context.
- **Variables of interest:** A variable of interest represents an attribute in the real world. The types and sources of sensor data used by data cleaning techniques vary from unique to heterogeneous sources.
- **Distributed:** In a distributed design, the cleaning tasks are performed locally in various components (e.g. sensor nodes) in contrast with the centralized design where the cleaning pipeline tasks are performed in one place (e.g. a server).
- **Fault Tolerance:** Sensor devices face many problems as it was described in 3.1 (e.g. node failure). Data cleaning approaches, which interact generally with the physical perception layer, should be resilient to faults and node failures in order to provide a stable service.
- **Confidence:** A confidence score represents how much trust we can put in an element. A confidence score could be calculated and attributed to different elements (e.g. dataset's values, sensor nodes).

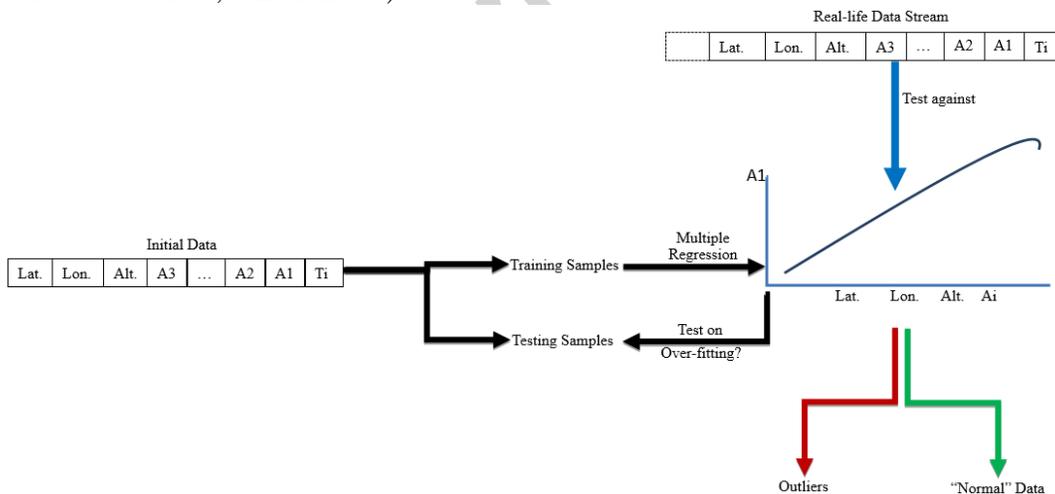


Fig. 15. Model-based data cleaning techniques process overview

6.5.4 Data cleaning techniques comparison

6.5.4.1 Model-based data cleaning techniques

Many of the characteristics (e.g. correlation) exhibited by the data gathered in the context of the IoT form the base for most of data cleaning model-based techniques. These techniques use many well-established mathematical models (e.g. regression

models) to perform data-related tasks, such as outlier detection. These mathematical models are surveyed in (Sathe et al., 2013).

- An in-network data cleaning approach for wireless sensor networks (Lei et al., 2016)

This paper proposes an in-network architecture for cleaning sensor data. The data cleaning process is composed of four steps, each of which is performed in a different physical component. The first stage is performed by individual sensor node. Each node checks its data using a built-in lightweight outlier detection based on the correlation of the measured attributes. The second stage consists of a cooperative process where neighbor nodes check unusual data for event detection. The third stage is performed in the sink to replace missing values with predicted ones. The fourth stage is computationally costly and is performed in the back-end server. It consists of providing regression parameters (e.g. using Gradient Descent method over recent historical data) for outlier detection algorithms running in sensor nodes and performing usual data mining tasks.

The outlier detection algorithm implemented in individual sensor node is using lightweight regression model and take advantage of the inherent correlation between various monitored attributes. For each reading, an abnormal-level value is used to label its degree of reliability. The more reliable a reading is, the less its abnormal-level will be. The event outlier detection algorithm is carried in the immediate neighbor node. This algorithm uses the Euclidian distance to measure the similarity between readings from neighbor nodes. If this distance is less than a user-specified thresholds, the abnormal-level is decreased to zero, thus, considering the suspicious reading as a legitimate event outlier. Otherwise, the abnormal-level is increased and the reading is considered faulty.

To measure the effectiveness of this technique, the authors implemented both outlier and event detection algorithms. A subset of sensor data from the Intel Lab scenario ("Intel Lab Data," 2004) was used for the experiments. The results reported by the authors showed a good accuracy for outlier detection and more energy-efficiency. However, our major concern is that the implementation was done in MATLAB and the algorithms were not running in a sensor node with constrained resources which might not give an exact measurement of the performance. Moreover, both stages 3 and 4 were only introduced without any further implementation or performance evaluation.

- Distributed Internal Anomaly Detection System for Internet-of-Things (Thanigaivelan et al., 2016)

The authors propose a distributed system for monitoring, detecting and blocking anomalous sensor nodes. Even though the proposed approach focuses on detecting nodes having an outlier behavior, it does indirectly enhance the quality of the generated data from the sensor network. In fact, anomalous nodes may produce erroneous data either intentionally (e.g. a hacked node) or unintentionally (e.g. fail-dirty node). Thus, blocking these anomalous nodes will improve the overall DQ.

The detection system delegates to each node the responsibility to monitor and grade its direct-neighbors (i.e. 1-hop neighbors) for any discrepancy in monitored features, with respect to the normal learnt behavior, such as suspicious packet size or data rate. If such anomalous behavior is detected, the node isolates and blocks the packets sent by its outlier neighbor, while also informing its parent about the incident.

The detection system has three subsystems built into the 6LoWPAN protocol stack and available for all individual nodes: (i) a Monitoring and Grading subsystem

(MGSS), (ii) a Reporting subsystem (RSS) and (iii) an Isolation subsystem (ISS). The MGSS handles behavior monitoring and grading-related operations. The RSS reports newly detected incident to the parent node through Distress Propagation Object messages (DPO), a novel control message integrated to the routing protocol for low-power and lossy networks (RPL). The ISS, giving the status of the neighbor, handles the tasks of allowing or discarding its packets. The three subsystems exchange information (e.g. the neighbor status) through a local repository. Giving their respective tasks, the MGSS and RSS operate in the Network Layer of the 6LoWPAN protocol stack because they need to access details of packets sent by the neighbor nodes, while the ISS operate in the Link Layer because it requires the ability to allow/discard transiting packets. It is worth noting that while the individual nodes are responsible for monitoring and reporting potential anomalous nodes, only the edge-router has the final call of whether to consider a node as anomalous or legitimate. To this end, the edge-router analyses the forwarded PDO messages from parent nodes for its final decision. Moreover, regardless of the edge-router final decision, the suspected node incurs an enforced isolation for a specified period of time. The edge-router oversees the changes in the network using network fingerprinting and performs periodic consistency checks through analyzing reports correlation.

The presented system's design suggests a number of advantages such as its scalability thanks to its distributed nature and energy and communication overheads minimization thanks to its reactive approach. However, no experimentation results nor performance analysis are reported in the paper.

- Anomaly Detection with HTM (Hole, 2016)

The paper proposes the use of the Hierarchical Temporal Memory (HTM) (Hawkins et al., 2011) to detect anomalies in streaming data. HTM is a machine learning algorithm modeled on how the neocortex performs the cognitive tasks such visual pattern recognition, understanding spoken language, etc. Among the basic functions performed by HTM are: learning, inference and prediction. In fact, HTM regions, representing the levels of the used hierarchical model, learn by discovering patterns on the input data using their spatial correlation (i.e. spatial patterns) and temporal correlation (i.e. temporal patterns or sequences) (e.g. identifying patterns in temperature and humidity readings produced in a certain area and how these patterns evolve over the span of each day). The nature of the input data themselves is not known for the HTM. The HTM's inference is done by comparing new input to previously learnt patterns. Finally, the prediction function compares current input data to the stored learnt sequences of patterns and tries to figure out which inputs are coming next.

For the outlier detection application, HTM computes an anomaly score for each pattern it receives. If the current received pattern is predicted (i.e. belonging to a well-known sequence of patterns), its anomaly score is set to zero. If the pattern is totally new (i.e. never encountered/learnt), its anomaly score is set to one. Otherwise, an anomaly score between zero and one is given to any partially predicted pattern. HTM suspects every new pattern that it never encountered. However, the more such pattern is encountered, the more normal it is considered. This is reflected in its anomaly score being decreased. When the HTM-based outlier detection starts operating, the ratio of flagged patterns will be high as the algorithm is still learning what it is a normal pattern (i.e. frequently encountered pattern) and what is not. Moreover, to handle the case of noisy data producing too many false positives alerts, HTM also uses an anomaly probability to identify how likely an obtained anomaly score is. The anomaly probabilities are computed over a window of previously calculated anomaly scores.

The authors presented an implementation for detecting anomalies in streaming metric data from running virtual machines clusters over the Amazon Web Services cloud platform. Another implementation is described for rogue behavior detection which aims to identify unusual and suspicious actions of human individuals in a monitored employment environment. The reported results shows that HTM could effectively detect anomalous patterns even when these are hard to find by a human monitor. While both implementation do not deal with data generated from smart things, the approach itself is usable for IoT giving its unsupervised learning features as noted by the authors.

- A Framework for Distributed Cleaning of Data Streams (Gill and Lee, 2015a)

A distributed framework for real-time environmental data streams cleaning is proposed. The presented Distributed Cleaning System (DCS) is designed as sub-system of a Stream Processing Engine (SPE). It is composed of a set of pipelined processes executed concurrently, over a cloud computing platform, for each incoming data stream. The same pipelined stages (Point, Smooth, Merge, Arbitrate and Virtualize) proposed by ESP (Jeffery et al., 2006a) are adopted by the DCS. However, their implementation are not necessarily done using a declarative language like in ESP but could be achieved with different kinds of statistical models. Moreover, the pipeline of functions in DCS is configurable such that certain functions could be implemented while others are not, depending on the application scenario's requirements. The proposed DCS architecture could be used to concurrently execute data cleaning task using various models for different applications.

The current implementation of DCS only features the Point stage performed using declarative cleaning queries and the Smooth stage performed using regression models (e.g. multiple regression models). Moreover, the different parts of the current DCS are separate. In fact, the Point stage is implemented in Streams-Esper (Scharrenbach, 2013) using its Event Processing Language (EPL) to remove obvious data outliers by imposing thresholds. Further, the regression models used in the Smooth stage are created and tested offline in R using the partially-cleaned data output of the Point stage. Finally, the actual Smooth stage is executed in a distributed framework using Spark Streaming (Zaharia et al., 2010). The trained models are imported as R objects and used to predict values on the incoming data stream.

The proposed approach for cleaning data streams has the advantage of being distributed which represents an important feature giving the needed resources required to process the large scale data generated in the context of IoT. However, the different parts of the system implementation should be integrated in order to further examine its performances.

- Context Aware Model-Based Cleaning of Data Streams (Gill and Lee, 2015b)

This paper proposes a context aware model-based technique for cleaning environmental sensor data. The novelty in this work is using context data from external sources to build the models. In fact, besides the environmental sensor data generated by the motes, geographical (e.g. location, elevation) and meteorological (e.g. wind speed) data are added to the training dataset while building the statistical models and also during the online cleaning of the environmental data stream. Most of these contextual data are obtained using web services.

The models were built using the Multiple Regression method. 70% of the initial data were used to build the models, while the remaining 30 % were used for

effectiveness testing and validation. Three types of models were built; linear, polynomial and Generalized Additive Models (GAM) for each variable of interest (i.e. the monitored pollutant). The best set of predictors was obtained using the Forward Stepwise Selection (FSS) technique. Similarly, to identify the best model, the authors used the Mean Squared Error (MSE) metric.

A two stage cleaning pipeline is implemented in a Data Cleaning System (DCS) architecture (Gill and Lee, 2015a). The first step is to clean obvious incorrect values (e.g. any temperature reading above 50 °C) using the Point stage process. In the second step (i.e. Smoothing stage), the different partially-cleaned tuples (enhanced with contextual data) are then passed to the prediction models. For each attribute of each tuple, the observed value is compared to the predicted value. The predicted value replaces the observed one any time the latter falls out of the 3 sigma rule interval used as the system error tolerance. During the cleaning of data stream, the MSE of each predictive model was calculated. The results shows that predictive models built on static data could also be effectively used for cleaning streaming data.

- An Outlier Detect Algorithm using Big Data Processing and Internet of Things Architecture (Souza and Amazonas, 2015)

A k-means clustering-based algorithm is used in conjunction with Big Data processing technologies and frameworks to detect outliers in huge sets of sensor data generated in the IoT.

The key elements of this approach are threefold. First, the k-means algorithm aims, starting from a set of observations (e.g. sensor data readings) and an initial set of k-clusters with their centroids, to iteratively group the observations into the available clusters using a distance function. In fact, in each iteration, each element is allocated to the cluster which has the shortest distance, giving the used distance function (e.g. Euclidean distance), between the cluster's centroid and the observation. Then, each centroid is updated as the mean of the within-cluster observations. The partitioning stops when the observations' allocation does not change, i.e. all clusters' centroids are stable. Second, a distributed computing architecture (Apache Hadoop) is adopted to handle the execution of the outlier detection process. In fact, the k-means clustering algorithm is executed concurrently on a distributed platform. Then, the output stable centroids and radii of each cluster are used as a model and are compared against each of the initial sensor readings. If the measured distance (e.g. Euclidean distance) between a sensor reading and each of the centroid is greater than the corresponding radius, the sensor reading is marked as an outlier. Third, the proposed distributed outlier detection module is integrated as a new layer in an IoT middleware.

The authors used the implementations of both the k-means algorithm and the Canopy clustering initialization algorithm provided in the Apache Mahout project ("Apache Mahout," 2014) which is built on top of the Apache Hadoop distributed computing project ("Apache Hadoop," 2014). Also, the outlier detection module is integrated as a new layer of the LinkSmart IoT middleware (Sarnovský et al., 2008). Finally, the reported experimentation results suggest that the outlier detection did well in identifying anomalous readings. However, the experiment only used raw data produced by a single sensor node. More experimentation involving multiple Hadoop instances and the LinkSmart middleware could give a clearer view on the performances of the detection algorithm.

- An Estimation Maximization Based Approach for Finding Reliable Sensors in Environmental Sensing (Zhang et al., 2015)

The authors propose a DQ enhancement approach based on selecting and relying only on reliable sensors in an environmental sensing deployment. Intuitively, the

more reliable a sensor is, the more confidence we can put on the readings it produces. Thus, by identifying faulty sensors and discarding their readings, the output data stream is cleaned.

An Expectation-Maximization algorithm (EM) is used in this approach. The main goal of such algorithm is to elaborate a statistical model (e.g. a model of environmental features) from observation data (e.g. environmental sensor data), given the existence of latent variables in the observed data (e.g. sensor faulty states). Both the statistical model and the latent variable are used to provide a likelihood function which the EM algorithm is set to maximize. Two steps are iteratively repeated until a convergence state is reached. First, an E-step which, giving a fixed statistical model, aims to find the latent variables. Second, an M-step which aims to find the new statistical model parameters that maximize the likelihood function giving fixed latent variables.

In their proposed approach, the authors initialize their algorithm with a linear model with Gaussian noises for the monitored environmental feature over a limited set of observations. The given justification is that within a limited timeframe, environmental phenomena tend to be characterized with smooth variations and spatio-temporal correlation. On the other hand, the used latent variable represent the faulty state of sensors. Initially, a domain-specific knowledge representing the probability of a reading being faulty is required. In each iteration, a selection array is elaborated from the current model and the readings reported by each sensor. It contains the decision to either take the readings of a sensor into account or to discard them. Then, according to the updated selection array, the statistical model is also updated to maximize a defined likelihood function. When the selection array converges, the algorithm stops.

The authors reported their experimentation findings on synthetic and real datasets. The results shows that EM approach effectively picks only non-faulty data and ignores the faulty ones especially when the readings produced by reliable sensor largely differs from the ones produced by faulty sensors. It also largely outperforms two other well-established data cleaning techniques.

- Vehicle Anomaly Detection based on Trajectory Data of ANPR System (Sun et al., 2015)

A trajectory-based outlier detection algorithm is proposed for identifying outlier vehicles using data generated by Automatic Number Plate Recognition systems (ANPR). The devised approach extracts temporal and spatial information from vehicles' trajectories in order to analyze and detect anomalous driving behavior (e.g. A wandering vehicle). The trajectories are constructed from the data provided by ANPR systems. In fact, each deployed video camera in the ANPR system monitors a specific area, referred to as a gateway, and automatically recognizes and reports vehicles driving through that area. The reported readings follow the model (time, gateway, license) denoting a specific vehicle (identified by its license plate), going through a specific area (i.e. gateway) at a specific time. More formally, the constructed trajectory is a set of readings reported by the ANPR cameras. To extract more significant insights about a vehicle's behavior, its trajectories are divided, using a time interval threshold, to a set of trips. The trips are then used as a source to extract a number of temporal and spatial features.

The extracted spatial features include the Route Length Factor (RLF) and the Route Rarity Factor (RRF). The RLF denotes a normalized value of the maximum path in a vehicle's trajectory. The RRF denotes a normalized value of the route rarity in a vehicle's trajectory. The route rarity represents how often a route is used, i.e.

how many vehicles have driven through that route. On the other hand, the extracted temporal features include the Activity of Daily Period (ADP) and the Activity of Specific Hours (ASH). The ADP denotes the activity of a vehicle on a daily-basis, i.e. on which days were the vehicle detected in the monitored area. The ASH denotes the activity of a vehicle on an hourly-basis in the monitored area.

The authors propose the use of two algorithms for detecting outlier vehicle: (i) a spatial outlier detection taking advantage of the extracted spatial features and (ii) a temporal outlier detection taking advantage of the extracted temporal features. The spatial outlier detection is based on the Cumulative Rotation Angles around the Centroid (CRAC) algorithm. Two phases are carried on during the spatial outlier detection. First, a list of candidate outlier vehicles are selected. A candidate vehicle is one whose RLF and RRF exceeds predefined thresholds, i.e. the vehicle takes long paths on uncommon routes. Second, for each candidate vehicle, a CRAC value is calculated for each of its trips. If any trip's calculated CRAC value exceeds a predefined threshold, the vehicle is flagged as being an outlier.

The temporal outlier detection uses a k-mean clustering algorithm and is performed over two steps. First, temporal features are extracted from the data of a subset of vehicles that exhibit normal behavior. Then, the k-mean algorithm is executed to group the extracted features (i.e. ADP and ASH) into k clusters representing the types of vehicles in the monitored area (e.g. trucks, buses, etc.). Second, for each vehicle, the minimum distance, using Euclidian distance, of its features from all the clusters' centroids is computed. If this minimum distance exceeds a predefined threshold, the vehicle is flagged as being an outlier.

The authors tested their approach both on synthetic and real datasets. The reported results show that both algorithms detects anomalous vehicles. However, they are both sensitive to the ANPR's recognition accuracy, especially the temporal outlier detection. Also, both algorithms require domain-specific knowledge and experimentation to specify various thresholds.

- Efficiently managing uncertain data in RFID sensor networks (Ma et al., 2014)

A framework for managing the uncertainty in RFID data in networked RFID systems for large scale object traceability is proposed. Two major components are described. A Markov-based global object tracking model and a local data management model. We focus on the latter in the remainder of this paragraph.

The data management model is executed locally in each node of the traceability network (e.g. a location in a supply chain network). Each node is responsible for managing data generated by its RFID readers and storing, in a local probabilistic database, a set of records following the proposed data model (time, tag ID, location, probability) denoting an object (i.e. tag ID), located in a certain node (i.e. location), at a certain time with a giving confidence (i.e. probability). Tracking applications could query these records in real-time to get the current position of an object, while tracing applications could use them to get an overview of the entire trajectory of such object.

Three sub-components compose the data management model: (i) a data processing component, (ii) a particle filter component and (iii) a data model component. In data processing stage, raw RFID readings are gathered and they are further represented as an observation variable Y (i.e. a set of n raw RFID readings reported by RFID readers). The location of an object, which is the only variable of interest in the object's state considered in this paper, is modeled with a continuous random variable X composed of a set of samples x_i paired with their probability p_i following a probability density function of the likelihood of the object being in a point x_i in a region covered by RFID readers. The sum of all probabilities of samples belonging to the same random variable equals 1, i.e. the object must be present in the traceability

network for each point in time within its transiting lifecycle. The probability p_i of an object being in a specific location x_i is inferred following Bayes' rule by computing a marginal posterior density of x_i over observed Y . This inference could be computationally costly as it requires to take into account all RFID readings captured thus far. In order to cope with this optimization problem, particle filtering, a sampling-based technique, is used to approximate this conditional distribution. In fact, instead of incorporating all the states of the objects, i.e. all locations, only a subset of these states, called particles, are used. The chosen particles are decided through a two-stage particle filter process which takes as input the whole set of an object's states (i.e. X) and a set of current observed readings (i.e. Y_t). First, a predicting step, outputs a set of candidate particles, representing a weighted version of the initial states. A list of qualified particles (i.e. candidate particles with the highest weights) are then selected for use in the approximation of the marginal posterior density. Second, an updating step, where particles are updated in function of the newly captured RFID readings and the previous states.

To evaluate their framework, the authors conducted a set of experiments in a simulated warehouse scenario. The presented results showed that the proposed approach succeeded in predicating the correct location of a moving pallet even though some readings were missing. Also, a comparison between the runtimes of the optimized (i.e. using particle filtering) and non-optimized approaches, showed that the first has better performance than the latter.

- Cleaning Environmental Sensing Data Streams Based on Individual Sensor Reliability (Zhang et al., 2014)

An incremental and reliability-based sensor data cleaning method is proposed. Instead of using the usual mean or median method, this approach incorporates the reliability of individual sensors into data cleaning, while incrementally adapting each sensor's reliability according to his performance (the quality of produced readings) in each data collection iteration.

The presented method has two main processes: (i) a reliability-based data cleaning, called Influence Mean Cleaning (IMC) and (ii) an incremental reliability update model. The IMC is a weighted mean for predicting true readings of a set of spatially correlated sensors. In fact, each reported reading is weighted by the reliability of its producing sensor. This way, readings produced by more reliable sensors, intuitively considered more accurate, will have greater impact on the predicted cleaned value than readings produced by less-reliable sensors. On the other hand, the reliability update model ensures that the reliability level affected to each individual sensor accurately describes its performances. At the end of each data collection and cleaning iteration, the reliability of each sensor is recalculated. This way, both the decline and improvement in a sensor's performance are captured and taken into account in the next cleaning iteration. In fact, a reward or penalty function is used to either increase the reliability of a sensor if its reported reading is within a tolerance threshold from the predicted value, or decreased otherwise. In other words, if a sensor reports readings that are in accordance with its neighbors, its reliability level increases. Otherwise, it is decreased.

This method does not require any knowledge on the technical specification of the used sensors to specify an initial reliability level. Instead, it sets an arbitrate reliability for each sensor and updates it with each iteration to reflect the true reliability of each sensor based on the readings it reports.

The authors reported experimentation results both on synthetic and real datasets. In both cases, the results show that the IMC outperforms both the median and the mean approaches, even though it does need some iterations to accommodate the reliability level of each sensor. It also demonstrates that the reliability update model effectively tracks the performance of each individual sensor both when it is declining and when it is recovering.

- Automated Sensor Verification using Outlier Detection in the Internet of Things (Javed and Wolf, 2012)

This technique starts from the observation that sensors generally monitor an attribute, typically smooth and continuous, in the real world. In order to apply the presented technique for outlier detection, a set of k variables of interest is considered. These variables are sensed using a set of n sensors deployed in a geographical area each of which is capable of sensing one or multiple variable of interest. The sensed state of a particular variable i at a given time t is represented by the set of values sensed by the sensors involved:

$$X_i(t) = \{X_{i1}(t), \dots, X_{in}(t)\} \quad (1)$$

No relationships are assumed between the sets of variables of interest (i.e., no domain specific expertise required) albeit they might exist. For all the variables, the sets of sensed values are considered as just sets of numbers. Furthermore, it is assumed that, since the sensed physical phenomenon is continuous, there is an existing underlying regularity. This regularity manifests in the form of spatial and temporal patterns or models. The combination of these two kinds of models makes it possible to determine the expected values for each variable of interest at any given spatial coordinates at any given instant of time.

This technique aims at automatically deriving a model for a variable of interest according to other variables of interest and spatial parameters (latitude, longitude, elevation) using multiple regression statistical modeling. During the process of deriving the underlying model in data, an n th degree polynomial is used to describe the relationship between data elements or variables. The greater the polynomial degree is, the better the formulated model is (i.e. there is a decrease in the error of data fitness versus the formulated model, which is measured using Standard Error and R-Squared). However, this does not indicate whether the model itself is “good” or not. In fact, a calculated model can fit greatly initial data (used to formulate the model) but it might not be the case for non-initial data, i.e. overfitting. To address this problem, datasets (initial values) are broken into Training Samples and Testing Samples. Training Samples are used to calculate the model and the Testing samples are used to test its efficiency. A “good” model is one that performs well both on Training Samples and Testing Samples (having an acceptable statistical errors vis-à-vis a defined threshold). The spatial model is calculated at each time step and used to detect outliers which could be caused by either a sudden drastic change in measures (e.g. a tornado, etc.) or erroneous reported sensed values by sensor(s). The derived model can also be used for spatial interpolation (e.g. to detect the outliers readings) or for temporal extrapolation (e.g. to predict sensor values in the near future).

This technique has shown promising results when applied in the case of a weather application. One major benefit is that no prior (domain-specific) knowledge is required to detect outliers. However, no further actions are taken after detecting the outlier. In fact, no means has been specified to decide on the real nature of the outlier. Also, this techniques uses multiple regression to define a variable of interest (e.g. pressure) based on other variables of interest (e.g. ambient temperature, etc.) and positional variables (latitude, longitude and elevation) which may pose problem when

there is only one monitored variable of interest. While it is true that a specific threshold is defined at the very beginning, the technique itself is fully automatic.

6.5.4.2 Declarative-based data cleaning techniques

- Towards Reusing Data Cleaning Knowledge (Almeida et al., 2015)

The paper presents a generic and domain-agnostic data cleaning methodology. It is based on the reuse of previously-specified cleaning rules across multiple data sources having potentially different data models and schemas. The generality and reusability of cleaning knowledge of the proposed approach come from the separation of abstract and concrete data cleaning operations and data source models.

The system architecture is composed of three layers; (i) an Abstract Data Layer (ADL), (ii) a Bridge Layer (BL) and (iii) a Concrete Data Layer (CDL). In order to clean data using this methodology, a domain expert specifies an abstract model of data cleaning operations using a domain-independent vocabulary and a set of references to a domain-specific conceptualization that abstracts the data sources to be cleaned. The resulting cleaning model is generic for all instances (i.e. data sources) of the used domain-specific conceptualization. Then, a set of transformations and mapping processes, contained in the BL, are executed to lower the abstraction level of the cleaning model specified in the ADL. In fact, a mapping process and two transformation processes are described. The mapping links concepts from the domain-specific conceptualization to concepts in the concrete data source when they have a different schema or model. The resulting mapping is used to transform data represented in different domain conceptualizations and also to rewrite the specified abstract data cleaning operations with respect to the concrete target data source. It is worth noting that the mapping and transformation modules could also be used to obtain an abstract domain conceptualization from a concrete data source model. Finally, the concrete data cleaning process is performed in the CDL by applying the rewritten data cleaning operations over data in the concrete data source.

One of the advantages of this approach is that, if any new data sources needs cleaning, only the mapping and transformation processes contained in the BL need to be updated. Once done, the data cleaning conceptual model defined by the domain expert could be used to obtain a new set of data cleaning operations for the new data source, i.e. reusing cleaning knowledge.

Even though the proposed approach seems to be viable for cleaning IoT data, the authors did not give any actual example or experiments of cleaning data streams. In fact, the only described hypothetical example scenario only dealt with static data. Moreover, no implementation is given even though the authors described the technologies that they envision to use in their system (e.g. RDF/OWL ontologies at the ADL).

- RFID Uncertain Data Cleaning Framework Based on Selection Mechanism (Xia et al., 2012)

The paper proposes a novel framework for RFID data cleaning which is supposed to reduce the time required to clean RFID datasets. In fact, the proposed architecture provides a selection mechanism where many potential data cleaning paths could be taken depending on the inherent uncertainty of each particular RFID reading. The authors presents the characteristics of uncertain RFID data as a key element in the design of the proposed approach.

The main components of this framework are: (i) a filter component, (ii) a pretreatment component, (iii) a stack of several optional data cleaning nodes and (iv) a sorting component. It is worth noting that the authors did not give much detail on the filter component besides being responsible of cleaning dirty data. The pretreatment component classifies the filtered data according to their tag number into processing units with a defined size. The stack of cleaning nodes contains a node for cleaning each of the major manifestations of uncertain RFID data (i.e. false readings, positive readings and redundant readings). Each processing unit is tested for a particular uncertainty problem (e.g. negative readings). If that processing unit data is affected with that particular problem, then that unit is passed through the corresponding cleaning node (e.g. negative readings cleaning node) to be cleaned, otherwise it passes on to the next node. Again, the authors did not give details on how this testing for all types of uncertainty problems is done. Moreover, no details are given on which actual data cleaning algorithms are implemented within each cleaning node. Finally, the cleaned processing unit are sorted in order to be streamed to the application layer.

The authors reported the results of testing their framework with simulated data. The results show that this framework is more time-efficient, compared to traditional RFID data cleaning frameworks, when the number of uncertain RFID data is significant.

- An Improved RFID Data Cleaning Algorithm Based on Sliding Window (L. Li et al., 2012)

The paper proposes an RFID data cleaning technique using a sliding window for non-uniform RFID data streams. This algorithm is an improvement of the SMURF approach (Jeffery et al., 2006b). The authors noted that the equation defining whether to change the window size or not, which is a key feature of the SMURF approach, is tightly related to the average reading rate of each reading cycle and that may cause problems. In fact, in the case of non-uniform RFID data streams, the average reading rate approach might not trigger changes in the window size when needed.

To improve the existing approach and overcome the challenges of non-uniform streams, the new approach suggests incorporating the average rate of the upcoming reading cycle as a factor of window size changing. Also, a threshold parameter is used to decide when to adapt the window size.

The authors analyzed the performance of their algorithm, the SMURF algorithm and a fixed-length sliding window algorithm. The results reported by the authors showed that their algorithm is more time-efficient compared to the SMURF algorithm. Moreover, while the SMURF and the improved algorithms performed fairly the same when processing stable RFID data stream, they both outperformed the fixed-length sliding window algorithm. Finally, when the RFID data stream is extremely unstable, the reported results show that the improved algorithm largely outperforms all the other algorithms.

6.5.5 Data cleaning techniques comparison overview

We summarize our comparison results in the following tables 5 and 6. It is worth noting that reference (1) corresponds to (Lei et al., 2016), (2) to (Thanigaivelan et al., 2016), (3) to (Hole, 2016), (4) to (Gill and Lee, 2015a), (5) to (Gill and Lee, 2015b), (6) to (Souza and Amazonas, 2015), (7) to (Zhang et al., 2015), (8) to (Sun et al., 2015), (9) to (Ma et al., 2014), (10) to (Zhang et al., 2014), (11) to (Javed and Wolf, 2012), (12) to (Almeida et al., 2015), (13) to (Xia et al., 2012) and (14) to (L. Li et al., 2012).

7. OPEN CHALLENGES AND FUTURE RESEARCH DIRECTIONS

According to our survey on DQ in IoT environments and existing data cleaning and outlier detection techniques for IoT data, it is clear that many challenges still need to be tackled in order to provide IoT-suited data cleaning infrastructure. In this section, we introduce these challenges. We also present some possible future directions for research that we believe can deliver efficient solutions and approaches for enhancing DQ in IoT environment.

7.1 Challenges

Ensuring DQ in the context of IoT still faces many challenging problems. In fact, most of the surveyed solutions lack the support of many features (as depicted in Tables 5 and 6) which are important in the context of IoT paradigm. The challenging issues caused by this design deficiency include:

- **Scalability:** IoT is expected to be deployed on a global scale with an unprecedented distributed aspect, even larger than the scale of the conventional Internet. However, most of the proposed solutions for data cleaning (Almeida et al., 2015; Hole, 2016; Javed and Wolf, 2012; L. Li et al., 2012; Sun et al., 2015; Xia et al., 2012; Zhang et al., 2015, 2014) are centralized which, in contrast of a distributed architecture, does not provide the needed flexibility and scalability for a large scale deployment.
- **Heterogeneity of data sources:** Data generated in the IoT come from different kind of “things” (e.g. sensors, RFID tags, etc.). Data cleaning techniques designed for IoT should be able to take into account heterogeneity of data sources especially WSN- and RFID-enabled data streams. Also, the proposed techniques should be able to handle different variables of interest to fulfill IoT applications' requirements which will likely provide complex services based on multiple parameters (e.g. adjust home temperature based on observed outer temperature, user habits, energy management, etc.).

Table 5. Data cleaning techniques comparison – Part 1

	Approach base		Scope		Data stream		Data characteristics		
	Model-based	Declarative-based	Outlier detection	Data cleaning	RFID	Sensor	Continuity	Spatial correlation	Temporal correlation
(1)	√			√		√		√	√
(2)	√		√			√		√	
(3)	√		√			√		√	√
(4)	√	√		√		√			
(5)	√	√		√		√		√	√
(6)	√		√			√		√	√
(7)	√			√		√	√	√	√
(8)	√		√			√		√	√
(9)	√			√	√				
(10)	√			√		√		√	

(11)	√		√			√	√	√	√
(12)		√		√	√	√			
(13)		√		√	√				
(14)		√		√	√				

Table 6. Data cleaning techniques comparison – Part 2

	Mode		Error/event separation	Multiple variables of interest	Automatic	Domain-agnostic	Distributed	Fault tolerance	Confidence
	Online	Offline							
(1)	√		√	√	√		√	√	√
(2)	√				√	√	√	√	√
(3)	√		√	√	√	√			√
(4)	√					√	√		
(5)	√			√			√		
(6)	√	√		√	√	√	√	√	
(7)		√			√				
(8)		√		√					
(9)	√	√		√			√		√
(10)	√	√			√				√
(11)	√			√	√	√			
(12)		√				√			
(13)	√								
(14)	√				√ (window size)				

- Domain-agnostic / automated verification: In the IoT vision, the “things” will communicate with one another autonomously to provide services based on their collaboration. A domain-agnostic approach for data cleaning will ensure that data transferred between the “things” are sound without the need for human intervention, which is crucial for a smooth creation of IoT services.
- Distributed architecture: Besides the scalability, a distributed architecture also provides fault tolerance and resilience to node failures. These features are important in the context of IoT as they allow the continuity and availability of data cleaning infrastructure that feeds ubiquitous services even when a failure occurs in a sub-system.

Moreover, another essential feature that lacks from most of the surveyed techniques is the ability to distinguish outliers representing errors from those representing events (Table 6’s “Error/event separation” column). As we have already mentioned, both events and errors could manifest as outliers. Lest of losing valuable knowledge, thus, potentially losing the capacity to act and react accurately, data cleaning techniques should be able to separate errors from events and accurately suppress only errors. In fact, in the context of IoT, where both the physical and the digital worlds are linked, being able to accurately determine the source of an outlier is particularly fundamental especially when acting on the physical world based on perceived phenomenon, and above all, when the system behavior involves critical tasks to be accomplished when such events occur (e.g. call emergency).

Further, most of the current surveyed techniques for data cleaning and outlier detection adopt a binary approach when it comes to classifying dataset's values i.e., a value is either an outlier or not an outlier (Table 6's "Confidence" column). Another approach would be to compute a confidence value for each dataset value (or for a group of values within a window, depending on the chosen granule), based on which an informed decision could be taken. The confidence approach would offer more flexibility for IoT applications to set their own confidence thresholds and to decide whether or not to accept or reject a particular element.

7.2 Future research directions

In this section, we present some possible future research directions we believe have the potential to enhance DQ in IoT:

- **IoT network traffic-based outlier detection:** Techniques based on network traffic analysis for anomaly and intrusion detection have already been successfully designed and used for the conventional internet (Giacinto and Roli, 2002; Li et al., 2006; Lu and Ghorbani, 2009; Münz et al., 2007; Shon et al., 2005; Zanero and Savaresi, 2004). However, the characteristics of the traffic generated and exchanged by the "things" in the IoT are still unknown (Borgia, 2014). Further researches on IoT traffic profiling are required in order to set the ground for an IoT traffic-based outlier detector. With this kind of approach in place, anomalous data packet, which do not conform to "normal" traffic patterns, sent by dysfunctional or malicious nodes (e.g. fail dirty nodes, hijacked nodes, etc.) could be detected.
- **Lightweight outlier detection techniques:** in the IoT, the "things" are mostly known for their scarce resources. For "things" to be able to self-control the quality of data they generate, there is clearly two options: 1) building more capabilities in smart things to be able to run existing data cleaning techniques on board, or 2) designing lightweight solutions that could be embedded in smart things. Implementing DQ control within "things" level would make the cleaning infrastructure capable of scaling and evolving with the same pace as IoT itself.
- **DQ assessment-based outlier detection:** All the outlier detection techniques we surveyed look for elements that differ from others without considering any other parameter (e.g. source sensor precision). DQ dimensions (e.g. preciseness, completeness, etc.) could be used as indicators of how good the received data are. An outlier detection approach that incorporates DQ assessment could reinforce decisions taken about the nature of dataset elements. Further, more research are needed to determine which DQ dimensions are the most relevant in the context of IoT data in order to achieve the best outlier detection rates.
- **Personalized DQ management platform:** DQ is subjective. Each data consumer has a unique vision of how "good" data should be depending on its core business and needs. In IoT, the number of pervasive applications (i.e. data consumers) increases each day and so does their DQ requirements. We believe that there is a need for an effective way to let each data consumer manages its DQ according to its own specifications and requirements without imposing too much constraints or overheads.
- **DQ management middleware:** Heterogeneity of data sources in IoT is unprecedented. Various types of smart things produce different kinds of data, each of which may or may not follow a pre-defined data model. In order to provide powerful ubiquitous services, pervasive applications tend to extract insights from data coming from various sources at once. Managing the quality of all these

different data is challenging. We need a solution to abstract all this heterogeneity in dealing with the quality of received data, so that the pervasive applications focus on delivering their services.

8. CONCLUSION

IoT promises a great potential by interconnecting millions of day-to-day objects in order to provide intelligent and ubiquitous services in favor of human beings. The amount of generated data from this global scale deployment is tremendous. The harvested data will serve as a base to extract insights about people, entities and phenomenon in order to provide IoT services. The quality of data is a major concern in this scenario. In fact, data trustworthiness is crucial for the user engagement and acceptance of the IoT paradigm. In this article, we surveyed DQ in the context of IoT. We identified data properties and their new lifecycle in the context of IoT. The concept of DQ is introduced and a set of generic and domain-specific DQ dimensions fit for use in assessing IoT data are selected. IoT-related factors endangering the quality of data are investigated. Further, an exhaustive qualitative analysis of their impact on various DQ dimensions, thus on the overall DQ, is presented. Moreover, we identified major DQ problems manifestation forms such as data outliers, multi-source data inconsistencies, etc. and we associated each manifestation class with its symptoms with respect to the affected DQ dimensions. We further studied data outliers as a major DQ problem and we investigated their impact in the context of IoT and its applications. Several techniques for enhancing DQ are presented. We focused on data cleaning techniques that promise to purify IoT data, which is vital for IoT acceptance. We reviewed and compared, using an extended taxonomy, these techniques to outline their characteristics and their fitness for use in the IoT. Finally, we discussed open challenges and possible future research directions we believe have the potential to set the ground for more efficient solutions and approaches for building IoT-suited cleaning infrastructures and enhancing DQ in the context of IoT environment.

IoT is a promising paradigm which has already yielded promising and exciting results. DQ plays a vital role in this context. Means for further DQ enhancement are to be further researched to ensure a wide deployment and acceptance of IoT.

ACKNOWLEDGMENT

The work of A. KARKOUCH leading to these results has received funding from the CNRST under the grant N° 1 8 U C A 2 0 1 5.

REFERENCES

- Aberer, K., Hauswirth, M., Salehi, A., 2006. Global Sensor Networks. *IEEE Commun. Mag.* 1–14.
- Abowd, G.D., Mynatt, E.D., 2000. Charting past, present, and future research in ubiquitous computing. *ACM Trans. Comput. Interact.* 7, 29–58. doi:10.1145/344949.344988
- Aggarwal, C.C., Ashish, N., Sheth, A., 2013. Chapter 12 THE INTERNET OF THINGS : A SURVEY FROM THE DATA-CENTRIC. *Manag. Min. Sens. Data* 383–428. doi:10.1007/978-1-4614-6309-2_12
- Aggarwal, C.C., Yu, P.S., 2008. A General Survey of Privacy-Preserving Data Mining Models and Algorithms. *Privacy-Preserving Data Min.* 11–52.
- Almeida, R., Maio, P., Oliveira, P., João, B., 2015. Towards Reusing Data Cleaning Knowledge. Springer International Publishing, pp. 143–150. doi:10.1007/978-3-319-16486-1_14
- Amadeo, M., Campolo, C., Molinaro, A., 2014. Multi-source data retrieval in IoT via named data networking. *Proc. 1st Int. Conf. Information-centric Netw. - INC '14* 67–76. doi:10.1145/2660129.2660148
- Apache Hadoop [WWW Document], 2014. URL hadoop.apache.org (accessed 7.10.16).
- Apache Mahout [WWW Document], 2014. URL <http://mahout.apache.org/> (accessed 7.10.16).
- Atzori, L., Iera, A., Morabito, G., 2010. The Internet of Things: A survey. *Comput. Networks* 54, 2787–

2805. doi:10.1016/j.comnet.2010.05.010

- Atzori, L., Iera, A., Morabito, G., Nitti, M., 2012. The social internet of things (SIoT) - When social networks meet the internet of things: Concept, architecture and network characterization. *Comput. Networks* 56, 3594–3608. doi:10.1016/j.comnet.2012.07.010
- Bailey, J.E., Pearson, S.W., 1983. Development of a Tool for Measuring and Analyzing Computer User Satisfaction. *Manage. Sci.* 29, 530–545. doi:10.1287/mnsc.29.5.530
- Ballou, D.P., Pazer, H.L., 2003. Modeling completeness versus consistency tradeoffs in information decision contexts. *IEEE Trans. Knowl. Data Eng.* 15, 240–243. doi:10.1109/TKDE.2003.1161595
- Ballou, D.P., Pazer, H.L., 1995. Designing Information Systems to Optimize the Accuracy-timeless Tradeoff. *Inf. Syst. Res.* 6, 51–72.
- Batini, C., Scannapieco, M., 2006. *Data Quality: Concepts, Methodologies and Techniques*, Springer. doi:10.1007/3-540-33173-5
- Bauer, M., Bui, N., Carrez, F., Giacomini, P., Haller, S., Ho, E., Jardak, C., Loof, J. De, Magerkurth, C., Nettsträter, A., Serbanati, A., Thoma, M., Walewski, J.W., Meissner, S., 2013. Introduction to the Architectural Reference Model for the Internet of Things.
- Bechhofer, S., 2009. OWL: Web ontology language, in: *Encyclopedia of Database Systems*. Springer, pp. 2008–2009.
- Bellavista, P., Corradi, A., Fanelli, M., Foschini, L., 2012. A survey of context data distribution for mobile ubiquitous systems. *ACM Comput. Surv.* 44, 1–45. doi:10.1145/2333112.2333119
- Borgia, E., 2014. The Internet of Things vision: Key features, applications and open issues. *Comput. Commun.* doi:10.1016/j.comcom.2014.09.008
- Branch, J.W., Giannella, C., Szymanski, B., Wolff, R., Kargupta, H., 2009. In-Network Outlier Detection in Wireless Sensor Networks. 26th IEEE Int. Conf. Distrib. Comput. Syst. ICDCS06 51–51. doi:10.1109/ICDCS.2006.49
- Burdakis, S., Deligiannakis, A., 2012. Detecting Outliers in Sensor Networks Using the Geometric Approach. 2012 IEEE 28th Int. Conf. Data Eng. 1108–1119. doi:10.1109/ICDE.2012.85
- Cardoso, T., Carreira, P., n.d. Overview on Energy Data Reporting.
- Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection. *ACM Comput. Surv.* 41, 1–58. doi:10.1145/1541880.1541882
- Chaplot, V., Darboux, F., Bourennane, H., Leguédois, S., Silvera, N., Phachomphon, K., 2006. Accuracy of interpolation techniques for the derivation of digital elevation models in relation to landform types and data density. *Geomorphology* 77, 126–141. doi:10.1016/j.geomorph.2005.12.010
- Dasu, T., Johnson, T., 2003. Exploratory data mining and data cleaning. *Comput. Math. with Appl.* 46, 980. doi:10.1016/S0898-1221(03)90170-2
- Dey, A.K., Abowd, G.D., 1999. Towards a Better Understanding of Context and Context-Awareness. *Comput. Syst.* 40, 304–307. doi:10.1007/3-540-48157-5_29
- Equille, L.B., 2007. Measuring and Modelling Data Quality for Quality-Awareness in Data Mining. *Quality* 126, 101–126. doi:10.1007/978-3-540-44918-8_5
- Erguler, I., 2015. A potential weakness in RFID-based Internet-of-things systems. *Pervasive Mob. Comput.* 20, 115–126. doi:10.1016/j.pmej.2014.11.001
- Evans, D., 2011. The Internet of Things - How the Next Evolution of the Internet is Changing Everything. CISCO white Pap. 1–11.
- Eysenbach, G., 2001. What is e-health? *J. Med. Internet Res.* 3, E20. doi:10.2196/jmir.3.2.e20
- Gallacher, S., Papadopoulou, E., Taylor, N.K., Williams, M.H., 2013. Learning user preferences for adaptive pervasive environments. *ACM Trans. Auton. Adapt. Syst.* 8, 1–26. doi:10.1145/2451248.2451253
- Geisler, S., Weber, S., Quix, C., 2011. ONTOLOGY-BASED DATA QUALITY FRAMEWORK FOR DATA STREAM APPLICATIONS 145–159.
- Giacinto, G., Roli, F., 2002. Intrusion detection in computer networks by multiple classifier systems. *Object Recognit. Support. by user Interact. Serv. Robot.* 2, 390–393. doi:10.1109/ICPR.2002.1048321
- Gill, S., Lee, B., 2015a. A framework for distributed cleaning of data streams. *Procedia Comput. Sci.* 52, 1186–1191. doi:10.1016/j.procs.2015.05.156
- Gill, S., Lee, B., 2015b. Context Aware Model-Based Cleaning of Data Streams 1–6.
- Gluhak, A., Krco, S., Nati, M., Pfisterer, D., Mitton, N., Razafindralambo, T., 2011. A survey on facilities for experimental internet of things research. *IEEE Commun. Mag.* 49, 58–67. doi:10.1109/MCOM.2011.6069710
- Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M., 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Futur. Gener. Comput. Syst.* 29, 1645–1660. doi:10.1016/j.future.2013.01.010
- Guo, B., Sun, L., Zhang, D., 2010. The architecture design of a cross-domain context management system. 2010 8th IEEE Int. Conf. Pervasive Comput. Commun. Work. PERCOM Work. 2010 499–504. doi:10.1109/PERCOMW.2010.5470618
- Guo, B., Zhang, D., Wang, Z., Yu, Z., Zhou, X., 2013. Opportunistic IoT: Exploring the harmonious

- interaction between human and the internet of things. *J. Netw. Comput. Appl.* 36, 1531–1539. doi:10.1016/j.jnca.2012.12.028
- Hand, D., Mannila, H., Smyth, P., 2001. Principles of data mining, Drug safety : an international journal of medical toxicology and drug experience. doi:10.2165/00002018-200730070-00010
- Hawkins, J., Ahmad, S., Dubinsky, D., 2011. Cortical Learning Algorithm and Hierarchical Temporal Memory. *Numenta Whitepaper* 1–68.
- Helfert, M., Foley, O., Ge, M., Cappiello, C., 2009. Analysing the Effect of Security on Information Quality Dimensions. 17th Conf. Inf. Syst. verona, italy, june 8-10 2785–2797.
- Hipp, J., Güntzer, U., Grimmer, U., 2001. Data Quality Mining-Making a Virute of Necessity., in: DMKD.
- Hofstra, N., Haylock, M., New, M., Jones, P., Frei, C., 2008. Comparison of six methods for the interpolation of daily, European climate data. *J. Geophys. Res. D Atmos.* 113, D21110. doi:10.1029/2008JD010100
- Hole, K.J., 2016. Anomaly Detection with HTM, in: *Anti-Fragile ICT Systems*. Springer International Publishing, Cham, pp. 125–132. doi:10.1007/978-3-319-30070-2_12
- Holler, J., Tsiatsis, V., Mulligan, C., Avesand, S., Karnouskos, S., Boyle, D., 2014. From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence.
- Hu, P., Indulska, J., Robinson, R., 2008. An autonomic context management system for pervasive computing. 6th Annu. IEEE Int. Conf. Pervasive Comput. Commun. PerCom 2008 213–223. doi:10.1109/PERCOM.2008.56
- Intel Lab Data [WWW Document], 2004. URL <http://db.csail.mit.edu/labdata/labdata.html> (accessed 5.3.15).
- Jardak, C., Walewski, J.W., 2013. Enabling Things to Talk. doi:10.1007/978-3-642-40403-0
- Javed, N., Wolf, T., 2012. Automated sensor verification using outlier detection in the Internet of things. Proc. - 32nd IEEE Int. Conf. Distrib. Comput. Syst. Work. ICDCSW 2012 291–296. doi:10.1109/ICDCSW.2012.78
- Jeffery, S.R., Alonso, G., Franklin, M.J., Hong, W., Widom, J., 2006a. Declarative support for sensor data cleaning. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* 3968 LNCS, 83–100. doi:10.1007/11748625_6
- Jeffery, S.R., Berkeley, U.C., Franklin, M.J., 2006b. Adaptive Cleaning for RFID Data Streams. *Vldb* 163–174.
- Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., Qiu, D., 2014. Security of the Internet of Things: perspectives and challenges. *Wirel. Networks* 20, 2481–2501. doi:10.1007/s11276-014-0761-7
- Kiritisis, D., 2011. Closed-loop PLM for intelligent products in the era of the Internet of things. *CAD Comput. Aided Des.* 43, 479–501. doi:10.1016/j.cad.2010.03.002
- Klein, A., Do, H.H., Hackenbroich, G., Karnstedt, M., Lehner, W., 2007. Representing data quality for streaming and static data. Proc. - Int. Conf. Data Eng. 3–10. doi:10.1109/ICDEW.2007.4400967
- Klein, A., Lehner, W., 2009b. Representing Data Quality in Sensor Data Streaming Environments. *J. Data Inf. Qual.* 1, 1–28. doi:10.1145/1577840.1577845
- Klein, A., Lehner, W., 2009a. How to Optimize the Quality of Sensor Data Streams. Proc. 2009 Fourth Int. Multi-Conference Comput. Glob. Inf. Technol. 00 13–19. doi:10.1109/ICCGL.2009.10
- Knox, E.M., Ng, R.T., 1998. Algorithms for Mining Datasets Outliers in Large Datasets. 24th Int. Conf. Very Large Data Bases 392–403.
- Kovatsch, M., Mayer, S., Ostermaier, B., 2012. Moving application logic from the firmware to the cloud: Towards the thin server architecture for the internet of things. Proc. - 6th Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Comput. IMIS 2012 751–756. doi:10.1109/IMIS.2012.104
- Lei, J., Bi, H., Xia, Y., Huang, J., Bae, H., 2016. An in-network data cleaning approach for wireless sensor networks. *Intell. Autom. Soft Comput.* 8587, 1–6. doi:10.1080/10798587.2016.1152769
- Li, F., Nastic, S., Dustdar, S., 2012. Data quality observation in pervasive environments. Proc. - 15th IEEE Int. Conf. Comput. Sci. Eng. CSE 2012 10th IEEE/IFIP Int. Conf. Embed. Ubiquitous Comput. EUC 2012 602–609. doi:10.1109/ICCSE.2012.88
- Li, L., Liu, T., Rong, X., Chen, J., Xu, X., 2012. An Improved RFID Data Cleaning Algorithm The Causes of the RFID Data Uncertainty. *Commun. Comput. Inf. Sci.* 312, 262–268. doi:10.1007/978-3-642-32427-7_36
- Li, S., Da Xu, L., Wang, X., 2013. Compressed sensing signal and data acquisition in wireless sensor networks and internet of things. *Ind. Informatics, IEEE Trans.* 9, 2177–2186.
- Li, X., Bian, F., Crovella, M., Diot, C., Govindan, R., Iannaccone, G., Lakhina, A., 2006. Detection and identification of network anomalies using sketch subspaces. Proc. 6th ACM SIGCOMM Internet Meas. - IMC '06 147. doi:10.1145/1177080.1177099
- Li, X., Lin, J., Li, J., Jin, B., 2015. A Video Deduplication Scheme with Privacy Preservation in IoT, in: *Computational Intelligence and Intelligent Systems*. Springer, pp. 409–417.
- Liu, L., Chi, L., 2002. Evolutional data quality: A theory-specific view. Proc. Seventh Int. Conf. Inf. Qual. 292–304.
- Lu, W., Ghorbani, A. a., 2009. Network anomaly detection based on wavelet analysis. *EURASIP J. Adv. Signal Process.* 2009. doi:10.1155/2009/837601
- Ma, H.D., 2011. Internet of things: Objectives and scientific challenges. *J. Comput. Sci. Technol.* 26, 919–

924. doi:10.1007/s11390-011-1189-5
- Ma, J., Sheng, Q.Z., Xie, D., Chuah, J.M., Qin, Y., 2014. Efficiently managing uncertain data in RFID sensor networks. *World Wide Web* 18, 819–844. doi:10.1007/s11280-014-0283-3
- Maletic, J., Marcus, a, 2000. Data Cleansing: Beyond Integrity Analysis. *Iq* 1–10.
- Mandagere, N., Zhou, P., Smith, M. a, Uttamchandani, S., 2008. Demystifying data deduplication. *Proc. ACM/FIP/USENIX Int. Middlew. Conf. companion Middlew. 08 Companion Companion* 08 12–17. doi:10.1145/1462735.1462739
- McNaull, J., Augusto, J.C., Mulvenna, M., McCullagh, P., 2012. Data and Information Quality Issues in Ambient Assisted Living Systems. *J. Data Inf. Qual.* 4, 4:1–4:15. doi:10.1145/2378016.2378020
- Mishra, N., Lin, C.C., Chang, H.T., 2015. A cognitive oriented framework for IoT big-data management prospective. *2014 IEEE Int. Conf. Commun. Probl. ICCP 2014* 124–127. doi:10.1109/ICCP.2014.7062233
- Münz, G., Li, S., Carle, G., 2007. Traffic anomaly detection using k-means clustering. *GI/ITG Work. MMBnet*.
- Nagib, A.M., Hamza, H.S., 2016. SIGHTED: A Framework for Semantic Integration of Heterogeneous Sensor Data on the Internet of Things. *Procedia Comput. Sci.* 83, 529–536. doi:10.1016/j.procs.2016.04.251
- Nataliia, L., Elena, F., 2015. Internet of Things as a Symbolic Resource of Power. *Procedia - Soc. Behav. Sci.* 166, 521–525. doi:10.1016/j.sbspro.2014.12.565
- National Intelligence Council, 2008. Disruptive Civil Technologies – Six Technologies with Potential Impacts on US Interests Out to 2025, Conference Report CR 2008-07.
- Naumann, F., 2002. Quality-Driven Query Answering for Integrated Information Systems, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/3-540-45921-9
- Nobles, A.L., Vilankar, K., Wu, H., Barnes, L.E., 2015. Evaluation of Data Quality of Multisite Electronic Health Record Data for Secondary Analysis. *Proc. 2015 IEEE Int. Conf. Big Data (Big Data)* 2612–2620. doi:10.1109/BigData.2015.7364060
- Otey, M.E., Ghoting, A., Parthasarathy, S., 2006. Fast distributed outlier detection in mixed-attribute data sets. *Data Min. Knowl. Discov.* 12, 203–228. doi:10.1007/s10618-005-0014-6
- Papadopoulos, G.Z., Beaudaux, J., Gallais, A., Noel, T., Schreiner, G., 2013. Adding value to WSN simulation using the IoT-LAB experimental platform. *Int. Conf. Wirel. Mob. Comput. Netw. Commun.* 485–490. doi:10.1109/WiMOB.2013.6673403
- Pascoe, J., 1998. Adding generic contextual capabilities to wearable computers. *Dig. Pap. Second Int. Symp. Wearable Comput. (Cat. No.98EX215)* 44. doi:10.1109/ISWC.1998.729534
- Perera, C., Member, C.H.L., Jayawardena, S., Chen, M., 2015. Context-aware Computing in the Internet of Things: A Survey on Internet of Things From Industrial Market Perspective 1–19. doi:10.1109/ACCESS.2015.2389854
- Perera, C., Member, S., Zaslavsky, A., Christen, P., 2014. Context Aware Computing for The Internet of Things : A Survey X, 1–41.
- Petrolo, R., Bonifacio, S.G., Loscri, V., Mitton, N., Petrolo, R., Bonifacio, S.G., Loscri, V., Mitton, N., Petrolo, R., Bonifacio, S.G., Loscri, V., Mitton, N., 2016. The discovery of " relevant " data-sources in a Smart City environment To cite this version : The discovery of " relevant " data-sources in a Smart City environment.
- Pinto-Valverde, J.M., Pérez-Guardado, M.Á., Gomez-Martinez, L., Corrales-Estrada, M., Lavariega-Jarquín, J.C., 2013. HDQM2: Healthcare Data Quality Maturity Model.
- Pujolle, G., 2006. An Autonomic-oriented Architecture for the Internet of Things. *IEEE John Vincent Atanasoff 2006 Int. Symp. Mod. Comput.* doi:10.1109/JVA.2006.6
- Qin, Y., Sheng, Q.Z., Curry, E., 2015. Matching Over Linked Data Streams in the Internet of Things. *IEEE Internet Comput.* 19, 21–27. doi:10.1109/MIC.2015.29
- Qin, Y., Sheng, Q.Z., Falkner, N.J.G., Dustdar, S., Wang, H., Vasilakos, A. V, 2014. When Things Matter: A Data-Centric View of the Internet of Things. *CoRR abs/1407.2*, 1–35.
- Rao, A.P., 2016. Quality Measures for Semantic Web Application. *Des. Solut. Improv. Website Qual. Eff.* 130.
- Rodríguez, C.C.G., Riveill, M., Antipolis, S., 2010. e-Health monitoring applications : What about Data Quality ?
- Rong, W., Xiong, Z., Cooper, D., Li, C., Sheng, H., 2014. Smart city architecture: A technology guide for implementation and design challenges. *China Commun.* 11, 56–69. doi:10.1109/CC.2014.6825259
- Said, O., Masud, M., 2013. Towards internet of things: Survey and future vision. *Int. J. Comput. Networks* 5, 1–17.
- Sarnovský, M., Kostelník, P., Butka, P., Hre\`v no, J., Lacková, D., 2008. First Demonstrator of HYDRA Middleware Architecture for Building Automation. *Znalosti* 2008 11.
- Sathe, S., Papaioannou, T.G., Jeung, H., Aberer, K., 2013. A survey of model-based sensor data acquisition and management, in: *Managing and Mining Sensor Data*. Springer, pp. 9–50.
- Scharrenbach, T., 2013. Streams-Esper [WWW Document]. URL

- <https://bitbucket.org/scharrenbach/streams-esper/wiki/Home> (accessed 7.8.16).
- Schilit, B., Adams, N., Want, R., 1994. Context-aware computing applications. *Work. Mob. Comput. Syst. Appl.* doi:10.1109/MCSA.1994.512740
- Sellitto, C., Burgess, S., Hawking, P., 2007. Information quality attributes associated with RFID-derived benefits in the retail supply chain. *Int. J. Retail Distrib. Manag.* 35, 69 – 87. doi:10.1108/09590550710722350
- Sethi, P., Kumar, P., 2014. Leveraging Hadoop Framework to develop Duplication Detector and analysis using MapReduce, Hive and Pig. *Proc. - 2014 7th Int. Conf. Contemp. Comput.* 454–460. doi:10.1109/IC3.2014.6897216
- Shah, M., 2016. Big Data and the Internet of Things, in: *Big Data Analysis: New Algorithms for a New Society*. Springer, pp. 207–237.
- Shanbhag, S., Wolf, T., 2008. Massively parallel anomaly detection in online network measurement. *Proc. - Int. Conf. Comput. Commun. Networks, ICCCN 261–266*. doi:10.1109/ICCCN.2008.ECP.63
- Shon, T.S.T., Kim, Y.K.Y., Lee, C.L.C., Moon, J.M.J., 2005. A machine learning framework for network anomaly detection using SVM and GA. *Proc. from Sixth Annu. IEEE SMC Inf. Assur. Work.* doi:10.1109/IAW.2005.1495950
- Sicari, S., Cappiello, C., De Pellegrini, F., Miorandi, D., Coen-Porisini, A., 2014. A security-and quality-aware system architecture for Internet of Things. *Inf. Syst. Front.* doi:10.1007/s10796-014-9538-x
- Sicari, S., Rizzardi, A., Grieco, L.A., Coen-Porisini, A., 2015. Security, privacy and trust in Internet of Things: The road ahead. *Comput. Networks* 76, 146–164. doi:10.1016/j.comnet.2014.11.008
- Singh, J., Pasquier, T., Bacon, J., Ko, H., Eysers, D., 2015. Twenty Cloud Security Considerations for Supporting the Internet of Things. *IEEE Internet Things J.* 4662, 1–1. doi:10.1109/JIOT.2015.2460333
- Soldatos, J., Kefalakis, N., Hauswirth, M., Serrano, M., Calbimonte, J.-P., Riahi, M., Aberer, K., Jayaraman, P.P., Zaslavsky, A., Žarko, I.P., others, 2015. Openiot: Open source internet-of-things in the cloud, in: *Interoperability and Open-Source Solutions for the Internet of Things*. Springer, pp. 13–25.
- Souza, A.M.C., Amazonas, J.R.A., 2015. An Outlier Detect Algorithm using Big Data Processing and Internet of Things Architecture. *Procedia Comput. Sci.* 52, 1010–1015. doi:10.1016/j.procs.2015.05.095
- Staab, S., Studer, R., 2007. Handbook on Ontologies. *Decis. Support Syst.* 654. doi:10.1007/978-3-540-92673-3
- Štěpánek, P., Zahradníček, P., Huth, R., 2011. Interpolation techniques used for data quality control and calculation of technical series: An example of a Central European daily time series. *Idojaras* 115, 87–98.
- Strong, D.M., Lee, Y.W., Wang, R.Y., 1997. Data quality in context. *Commun. ACM* 40, 103–110. doi:10.1145/253769.253804
- Sun, Y., Zhu, H., Liao, Y., Sun, L., 2015. Vehicle Anomaly Detection Based on Trajectory Data of ANPR System. 2015 IEEE Glob. Commun. Conf. 1–6. doi:10.1109/GLOCOM.2015.7417520
- Sundmaeker, H., Guillemin, P., Friess, P., 2010. Vision and challenges for realising the Internet of Things, ... the Internet of Things. doi:10.2759/26127
- Tanganelli, G., Mingozzi, E., Vallati, C., 2013. A Distributed Architecture for Discovery and Access in the Internet of Things.
- Thanigaivelan, N.K., Kanth, R.K., Virtanen, S., Isoaho, J., 2016. Distributed Internal Anomaly Detection System for Internet-of-Things. 2016 13th IEEE Annu. Consum. Commun. Netw. Conf. 0–1.
- Tsai, C.W., Lai, C.F., Chiang, M.C., Yang, L.T., 2014. Data mining for internet of things: A survey. *IEEE Commun. Surv. Tutorials* 16, 77–97. doi:10.1109/SURV.2013.103013.00206
- Uckelmann, D., Harrison, M., Michahelles, F., 2011. Architecting the Internet of Things.
- Ukil, A., Sen, J., Koilakonda, S., 2011. Embedded security for internet of things. *Proc. - 2011 2nd Natl. Conf. Emerg. Trends Appl. Comput. Sci. NCETACS-2011* 50–55. doi:10.1109/NCETACS.2011.5751382
- Vajda, V., Furdík, K., Glova, J., Sabol, T., 2011. The EBBITS Project : An Interoperability The EBBITS Project : An Interoperability platform for a Real-world Internet of Things domain of platform for a populated populated Internet Things domain 309–312.
- van der Togt, R., Bakker, P.J.M., Jaspers, M.W.M., 2011. A framework for performance and data quality assessment of Radio Frequency Identification (RFID) systems in health care settings. *J. Biomed. Inform.* 44, 372–383. doi:10.1016/j.jbi.2010.12.004
- Vongsingthong, S., Smachat, S., 2015. A Review of Data Management in Internet of Things. *KKU Res. J.* 20, 215–240.
- Wang, C.D., Mo, X.L., Wang, H. Bin, 2009. An intelligent home middleware system based on context-Awareness. 5th Int. Conf. Nat. Comput. ICNC 2009 5, 165–169. doi:10.1109/ICNC.2009.566
- Wang, J., 2016. Data Cleaning : Overview and Emerging Challenges 16–21.
- Wang, R., Wang, Z., Lian, D., 2012. A study of the unification of multisensor data. *ICALIP 2012 - 2012 Int. Conf. Audio, Lang. Image Process. Proc.* 805–810. doi:10.1109/ICALIP.2012.6376724
- Wang, R.W., Strong, D.M., 1996. Beyond Accuracy: What Data Quality Means to Data Consumers. *J.*

- Manag. Inf. Syst. 12, 5. doi:10.2307/40398176
- Weinberg, J., 2004. RFID and Privacy. SSRN Electron. J. doi:10.2139/ssrn.611625
- Wickramasuriya, J., Datt, M., Mehrotra, S., Venkatasubramanian, N., 2004. Privacy protecting data collection in media spaces. Proc. 12th Annu. ACM Int. Conf. Multimed. - Multimed. '04 48. doi:10.1145/1027527.1027537
- Wlodarczak, P., Soar, J., Ally, M., 2016. Inclusive Smart Cities and Digital Health 9677, 321–331. doi:10.1007/978-3-319-39601-9
- Xia, X., Xuan, L., Li, X., Li, Y., 2012. RFID uncertain data cleaning framework based on selection mechanism 312 CCIS, 234.
- Yan, Z., Wang, M., Li, Y., 2016. Encrypted Data Management with Deduplication in Cloud Computing, in: IEEE Cloud Computing. pp. 28 – 35. doi:10.1109/MCC.2016.29
- Yan, Z., Zhang, P., Vasilakos, A. V., 2014. A survey on trust management for Internet of Things. J. Netw. Comput. Appl. 42, 120–134. doi:10.1016/j.jnca.2014.01.014
- Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I., 2010. Spark : Cluster Computing with Working Sets. HotCloud'10 Proc. 2nd USENIX Conf. Hot Top. cloud Comput. 10. doi:10.1007/s00256-009-0861-0
- Zanero, S., Savaresi, S.M., 2004. Unsupervised learning techniques for an intrusion detection system. Proc. 2004 ACM Symp. Appl. Comput. - SAC '04 412. doi:10.1145/967900.967988
- Zeng, D., Guo, S., Cheng, Z., 2011. The web of things: A survey. J. Commun. 6, 424–438. doi:10.4304/jcm.6.6.424-438
- Zhang, Y., Szabo, C., Sheng, Q.Z., 2015. An Estimation Maximization Based Approach for Finding Reliable Sensors in Environmental Sensing, in: 2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS). IEEE, pp. 190–197. doi:10.1109/ICPADS.2015.32
- Zhang, Y., Szabo, C., Sheng, Q.Z., 2014. Cleaning Environmental Sensing Data Streams Based on Individual Sensor Reliability. Springer International Publishing, pp. 405–414. doi:10.1007/978-3-319-11746-1_29
- Zhang, Y.Z.Y., Meratnia, N., Havinga, P., 2010. Outlier Detection Techniques for Wireless Sensor Networks: A Survey. IEEE Commun. Surv. Tutorials 12, 1–12. doi:10.1109/SURV.2010.021510.00088
- Zhao, K., Ge, L., 2013. A survey on the internet of things security. Proc. - 9th Int. Conf. Comput. Intell. Secur. CIS 2013 663–667. doi:10.1109/CIS.2013.145
- Zheng, Y., Zhou, X. (Eds.), 2011. Computing with Spatial Trajectories. Springer New York, New York, NY. doi:10.1007/978-1-4614-1629-6

