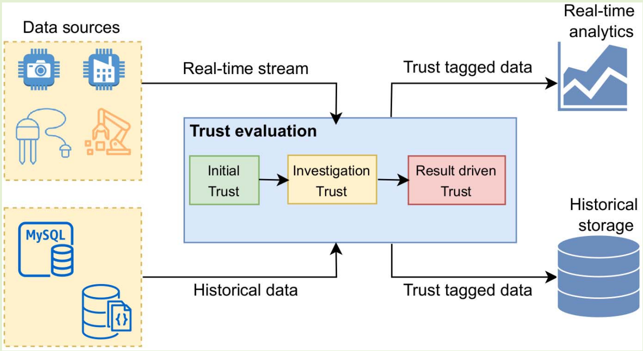
End-to-End Data Quality Assessment Using Trust for Data Shared IoT Deployments

John Byabazaire, Gregory M.P. O'Hare, *Member, IEEE*, and Declan T. Delaney



***Abstract*—Continued development of communication technologies has led to widespread Internet-of-Things (IoT) integration into various domains, including health, manufacturing, automotive, and precision agriculture. This has further led to the increased sharing of data among such domains to foster innovation. Most of these IoT deployments, however, are based on heterogeneous, pervasive sensors, which can lead to quality issues in the recorded data. This can lead to sharing of inaccurate or inconsistent data. There is a significant need to assess the quality of the collected data, should it be shared with multiple application domains, as inconsistencies in the data could have financial or health ramifications. This article builds on the recent research on trust metrics and presents a framework to integrate such metrics into the IoT data cycle for real-time data quality assessment. Critically, this article adopts a mechanism to facilitate end-user parameterization of a trust metric tailoring its use in the framework. Trust is a well-established metric that has been used to determine the validity of a piece or source of data in crowd-sourced or other unreliable data collection techniques such as that in IoT. The article further discusses how the trust-based framework eliminates the requirement for a gold standard and provides visibility into data quality assessment throughout the big data model. To qualify the use of trust as a measure of quality, an experiment is conducted using data collected from an IoT deployment of sensors to measure air quality in which low-cost sensors were colocated with a gold standard reference sensor. The calculated trust metric is compared with two well-understood metrics for data quality, root mean square error (RMSE), and mean absolute error (MAE). A strong correlation between the trust metric and the comparison metrics shows that trust may be used as an indicative quality metric for data quality. The metric incorporates the additional benefit of its ability for use in low context scenarios, as opposed to RMSE and MAE, which require a reference for comparison.**

***Index Terms*—Big data model, data quality, Internet of Things (IoT), machine learning, trust.**

I. Introduction

THE Internet-of-Things (IoT) paradigm has seen tremendous growth in the industry in the last five years. The number of connected devices in various sectors has also grown. This has, in turn, led to an increase in the amount of data generated and consumed. This exponential increase in data collected and consumed led to the IoT big data wave. This is characterized by volume, velocity, variety, veracity, and value, the 5V's of big data as they are known [1]. As this data is collected, it must undergo several stages from collection to decision-making. These stages form the big data model.

The big data model is a series of stages that the data must undergo from when it is created to when it is used. Each preceding stage is critical for the success of the next stage. Fig. 1 shows the various stages of the big data model. Data collection, data preprocessing, data processing, and data use are separate stages of the big data model. It is beneficial to interrogate data quality independently at each stage in the model. It can also be argued, however, that data quality should be reviewed longitudinally through the model for a given input and use case. For each stage, data can have different properties, and therefore, data quality has to be assessed separately but also represented differently. This is equally true for different data users and applications within the IoT ecosystem.

The data generated and consumed within IoT comes from several domains including, but not limited, to: 1) smart homes; 2) smart cities; 3) manufacturing; and 4) automotive

Manuscript received 10 August 2022; accepted 30 August 2022. Date of publication 9 September 2022; date of current version 14 October 2022. This work was supported in part by the Science Fund Ireland (SFI) Strategic Partnership Program under Grant 16/SPP/3296 and in part by Origin Enterprises PLC. The associate editor coordinating the review of this article and approving it for publication was Dr. Prosanta Gope. *(Corresponding author: John Byabazaire.)*

John Byabazaire is with the School of Computer Science, University College Dublin, Dublin 4, D04 V1W8 Ireland (e-mail: john.byabazaire@ucdconnect.ie).

Gregory M.P. O'Hare is with the School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, D02 PN40 Ireland, and also with the School of Computer Science, University College Dublin, Belfield, Dublin 4, D04 V1W8 Ireland (e-mail: gregory.ohare@tcd.ie).

Declan T. Delaney is with the School of Electrical and Electronic Engineering, University College Dublin, Dublin 4, D04 V1W8 Ireland (e-mail: declan.delaney@ucd.ie).

Digital Object Identifier 10.1109/JSEN.2022.3203853

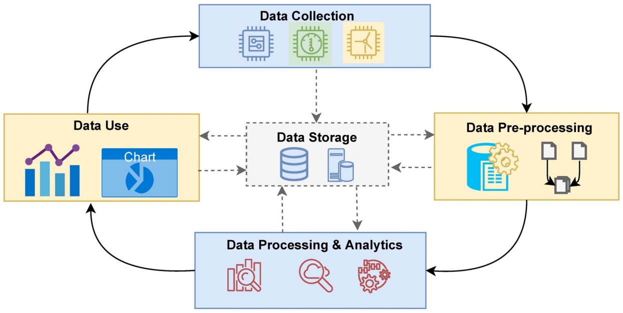


Fig. 1. Big data model.

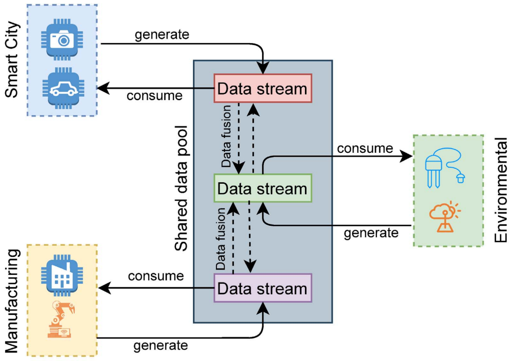


Fig. 2. Shared IoT ecosystem.

to environmental sensing. Current discussions introduce the opportunities that sharing and consuming data across such domains of the IoT ecosystem would have for further innovation in the IoT space. This sharing of data across multiple domain spaces is referred to as data-shared IoT [2]. The example in Fig. 2 shows how a smart city application can benefit from data fusion of its own data with data from other IoT applications for better insights. For this fusion to be fruitful, it is important to ensure that the shared data conform to certain quality standards and can be trusted by the consuming application.

This article presents a mechanism that aims to achieve the following objectives: 1) tailor assessment as per user requirements; 2) decouple metrics from the evaluation strategy; and 3) allow for longitudinal fusion of quality scores. Achieving these aims comes with challenges associated with them. Here, the aims are explained, and in Section IV, the challenges are highlighted.

Decoupling the mechanics of providing a quality score from the means of evaluating the quality. The big data cycle (BDC) has various stages that data must undergo. At each stage, the data can have different quality issues. To determine and address data quality issues, it is beneficial to assess quality at each stage considering only the data properties at that stage. The ability to map data quality to the individual stage of the BDC will be referred to as the mechanics of providing or advertising quality. This can be separated from the metric used to evaluate the quality. This article uses a trust metric to provide a quality score, however, the mechanism can be applied to other quality metrics. This is illustrated in Section VII where each trust stage is decoupled and integrated into the BDC.

This separation is useful to achieve a domain and use-case-agnostic solution. Each unique domain and use case can define its own metric for calculating a quality score without affecting the mechanics of applying quality assessment or advertising quality. Furthermore, when the quality requirements of an application change, the metric of assessing quality can be customized to that application's needs without affecting, or needing to change the underlying mechanics of assessment.

Longitudinal fusion of data quality defines a means to combine the various quality scores from each stage of the BDC into a single score that is representative of the end-to-end processes that the data has undergone. This process should be independent of the metric of assessing quality. Any fusion technique can be used here. In Section VIII, a naive fusion approach was used to distinguish between a low-quality stream and a gold reference stream. The intention, however, is to investigate more advanced fusion techniques.

A tangible link between data quality, data quality types, and their effect on data through the stages of the BDC has been demonstrated [2]. This concept, however, is yet to be implemented. This describes the longitudinal relationship between the various data quality issues across the BDC. For example, knowing which data quality issues are present in the initial stages of the BDC (data preprocessing) and how this affects the next stage can help determine a data-processing technique in that stage.

Data quality tailoring allows an application to customize quality assessment to suit its requirements. Quality itself is subjective and so should be evaluated as such. Data that is good for a particular application or use case might not be for the other. Each application should be able to define its own quality. For example, if data accuracy is important for a particular application and data latency (timeliness) is not, then the quality score should be customized to reflect that. This is illustrated by the use case in Section X, based on a custom data quality score, and each application can connect or disconnect from a data source.

Current solutions aim to optimize quality assessment for a given use case [3], [4], [5], [6]. The resulting solutions, however, may not be applicable to another use case. Taleb *et al.* [7] described the importance of integrating quality assessment with the BDC, connecting quality assessment with the source of the quality issues. This approach, however, has not been implemented.

This article presents a real-time end-to-end implementation of a data quality assessment framework that can be used to assess the quality of data in IoT deployments. The framework leverages trust as a means to assess quality where no reference or ground-truth data is available. Assessment of quality throughout the BDC allows identification of introduced quality issues at each stage. The framework is agnostic of the trust metric or fusion technique used; however, a trust metric presented in earlier work [2] is used in the implementation and testing.

The rest of the article is structured as follows: Section II presents background information. Section III presents the state-of-the-art. Section IV details the existing challenges in implementing quality assessment in IoT environments. Section V presents the motivation for the proposed framework and highlights why trust is an important metric for the assessment of data quality in IoT. Section VI provides a detailed description of the framework including the mathematical formulation of the framework and microservice-based implementation of the solution. Section IX presents the testing environment, datasets employed, evaluation strategy, and results of the evaluation. Finally, Section XI presents the concluding remarks and future work.

II. Background

This section introduces three concepts that are central to this work: 1) data quality; 2) data quality dimensions (DQDs) and trust; and 3) their application within an IoT context. It then highlights the use of trust as a measure of quality.

A. Data Quality

Data collected from sensing IoT devices is of paramount importance today. Such data is being used to advance innovations and inform decision-making. Much of this data comes from low-cost sensor devices, which are inherently unreliable [8]. Assessing and ascertaining the quality of such data before using it is therefore important. Data quality has widely been studied in database management [9], [10], [11] and also in the big data context [12]. Poor data quality management can have adverse negative effects on business decisions [13].

Data quality is subjective, making it dependent on the use case and domain area. It has been defined differently in academic and industrial contexts [14]. Sidi *et al.* [3] defined data quality based on how appropriate it is for use based on user need. According to Heravizadeh *et al.* [15], quality means the totality of the characteristics of an entity (data) that bear on its ability to satisfy stated and implied needs.

B. Data Quality Dimensions

DQDs provide an acceptable way to measure data quality. Several authors have defined different DQD, each with associated metrics [14]. A DQD is a characteristic or feature of information for classifying information and data requirements. As such, it offers a way for measuring and managing data quality as well as information [3]. It is important to note that there is no standard definition of DQD that is acceptable as domain-independent [16]. It is argued that some of these could be task-independent, and therefore not restrained by the context of application while others are task-dependent [17]. In Lee *et al.* [18], many of these were studied and later summarized them into four main categories as shown in Table I.

C. Trust

Trust can be defined as the belief of a trustor in a trustee that the trustee will accomplish a given task by satisfying the trustor's expectation [16]. Different users have different requirements that must be satisfied before they can trust a source. Examples of these can relate to DQDs including reliability, competence, credentials, and reputation.

TABLE I
DQD Categories

Data Quality category	Data Quality dimensions
Intrinsic	Accuracy, Objectivity, Believability, Reputation
Accessibility	Accessibility, Access security
Contextual	Relevancy, Value-added,Timeliness, Completeness, Amount of data
Representational	Interpretability, Ease of understanding, Concise representation, Consistent representation

In the era where information is widely available, users are tasked with gauging the quality of such. From the trust perspective, a data source can build a reputation over time to become trustworthy. Trust in itself is a process, and therefore trust can be formed, improved, and also lost. Some data sources have built trust over time and now they are more trusted than others. Trust has been widely used in other areas. In service computing, Malik and Bouguettaya [19] and Chang *et al.* [20] used trust to select the best service for a user. Jøsang *et al.* [21] proposed systems that could be used to derive measures of trust and reputation for Internet transactions.

D. Trust as a Measure of Data Quality

In a wider sense, trust has been used as a measure of quality, especially in information systems. It is assumed that if more people trust a product or service, it has better quality and vice versa. This same principle has been used widely in information search on the Internet and more recently in recommender engines [22].

Like trust, quality is an iterative process that must be constantly reassessed. To achieve a level of trustworthiness, different trust attributes must be evaluated at every stage and how these contribute to each other. The uniqueness of trust as a metric for data quality assessment lies in its properties. Byabazaire *et al.* [23] highlights these properties and how each can be used to harness trust as data quality metrics, especially in data-shared IoT scenarios. For example, trust is personalizable. In IoT, each application has a unique description of data quality.

III. State-of-the-Art

Several solutions have been proposed to help ensure that data retain its quality within database management and a few in the context of IoT and big data. While data quality assessment in a general context can be considered a mature field of study, data quality assessment in the context of IoT has not yet been fully explored [24]. There are several methods to ensure data retains its quality. This section contrasts two approaches to data quality assessment that relate to this work. The first approach aims to develop DQDs that can be used by domain experts to assess data quality both generally, and in the context of IoT. The other approach aims to take these DQDs and develop solutions that automate the process of assessing and improving data quality.

TABLE II
COMPARISON OF THE VARIOUS EVALUATION STRATEGIES

Research work	Evaluation strategy		
	Domain expert knowledge	Unique process	General evaluation
A product perspective on total data quality management [26]	✓	×	×
Aimq: a methodology for information quality assessment [32]	✓	×	×
Automated sensor verification using outlier detection in the Internet of thing [35]	×	✓	×
Data filtering system to avoid total data distortion in IoT networking [36]	×	✓	×
The daquincis architecture: a platform for exchanging and improving data quality in cooperative information systems [33]	✓	✓	×
Hybrid Information Quality Management (HIQM) methodology [34]	✓	×	×
Evaluating Sensor Data Quality in Internet of Things Smart Agriculture Applications [26]	×	✓	×
Proposed system	×	×	✓

A. Development of DQDs

DQDs offer a way for measuring and managing data quality as well as information [25]. More precisely, DQDs describe the various measurable metrics of data quality. The main aim of the solutions is to define new DQDs and evaluate them based on expert opinion.

The data quality assessments framework based on DQDs date back to 1998. Total Data Quality Management (TDQM) by Wang [27] has been widely accepted within database management systems [28], [29] and within big data and IoT systems [30], [31], [32]. One of its core advantages is that it emphasizes an iterative approach to data quality management. Data users specify their requirements, and data engineers (information product engineers) translate these into DQDs that are measurable. Finally, expert knowledge is used to validate the requirements against the output of the engineers. Their implementation proposes 15 DQDs that can be applied to various domains and have been widely adopted. The evaluation of the framework is based on domain expert knowledge.

Lee *et al.* [18] later proposed AIM quality (AIMQ) that is based on TDQM. The major contribution of this work classifies DQDs into four categories: 1) intrinsic; 2) contextual; 3) representational; and 4) accessibility. Several other methodologies have also been proposed based on TDQM.

While the solutions above are more general, more recently, Alrae *et al.* [33] proposed "House of Information Quality framework for IoT systems." This differs from the above solutions in that it compares DQDs associated with information quality and the core IoT elements to define DQDs that are necessary for IoT applications. Like the above solutions, expert opinion is used as a significant validation method. This is a good validation strategy, but not good for runtime assessment.

B. Application of DQDs

This section highlights solutions that use DQDs to implement applications for assessing and improving that quality. In these, some have advanced a data-centric approach by trying to mitigate the errors in the data itself [34], [35], [36], and others have proposed a process-centric approach where the data collection process is assessed [3], [4], [5], [6].

Javed and Wolf [36] presented a technique that leverages spatial and temporal interpolation to identify outliers in sensor reading. They evaluate their solution using weather sensing and conclude that the same method can be used in any application domain where the underlying phenomenon is continuous.

Tsai *et al.* [34] proposed an abnormal sensor detection architecture that leverages machine-learning techniques, using a trained Bayesian model that can predict values of sensor nodes via other correlated sensors. Their results show they can detect abnormal sensors in real-time.

Vilenski *et al.* [35] looked at a multivariate anomaly detection technique for ensuring the data quality of dendrometer sensor networks. The anomalous sensors are identified statistically by comparing a sensor's readings to an expected reading from a similar, healthy sensor network. As a gold standard, expert knowledge was used to assess the system. The above solutions address a single DQD (accuracy) by employing anomaly detection techniques.

Contrary to the above, Kim *et al.* [37] used accurate and consistent DQDs and proposed a system for filtering data based on the sensing objects. By employing a Bayesian classifier, the classifier can filter sensing objects with inaccurate data and then deliver data with integrity to the server for analysis. The performance of the proposed data-filtering system is evaluated through computer simulation.

This research identifies the necessity of addressing multiple DQDs and directs the work in this article for a general framework to apply multiple DQDs in assessment, with those DQDs specified by the end user.

Current approaches are evaluated by either expert knowledge (Section III-A), by a unique process (Section III-B), or by bespoke gold standard reference measure for a given use case. The evaluation strategy is thus specific to the use case. Identifying a means to evaluate data quality assessment strategy via a benchmark remains an open challenge. Table II compares some of the previous research and the evaluation strategies used against the proposed approach.

IV. EXISTING CHALLENGES

The approach to data quality assessment in this article is to investigate how the aspects of data quality affect the performance of each stage in the big data model and how this affects subsequent stages.

To understand how data quality assessment proliferates and affects data use cases, it is essential to understand the relationship between data quality and the big data model. Thus far, the literature does not consider data quality as a fundamental aspect of the big data model. A challenge exists with regard to structuring DQDs within the big data model so that the effect of data quality is identified throughout the model. Currently, it is difficult to know which part of the big data model is responsible for what portion of the overall data quality score. These are referred to as structure-related challenges [2].

With a given data quality structure in mind, considerations on how data quality measurements from one stage of the big data model can and should affect data quality measurements at other stages in the big data model. This may involve combining or weighting quality measurements for a given stage or use case. A number of challenges exist in this space and are referred to as method-related challenges.

Implementing a given assessment structure and methodology into a data pipeline from end to end considering the uniqueness of each stage of the BDC remains a challenge. These are referred to as implementation challenges.

A. Method-Related Challenges

1) Data quality can be highly subjective. A single data point or source can have varying qualities depending on the use-case context. How might data quality be represented in a general manner throughout the big data model yet allow subjective assessment by two (or more) end users?
2) Data quality is measured and represented in different forms depending on the stage and context within the big data model. How can these data quality measures be combined across the big data model stages to infer a quality metric which is useful for use-case quality determination? This article addresses these challenges by providing a means of decoupling the mechanics of providing a quality score and the methods of evaluating quality.

B. Implementation-Related Challenges

1) The current implementation of data collection/processing solutions for IoT and big data are based on a data pipeline. The BDC is broken down into a set of defined individual and independent services. For a data assessment solution to be feasible, it should be integrated into such data pipelines. Challenges exist in choosing the significant stages in the pipeline where data should be evaluated.
2) Other challenges include, fusing different scores from independent stages of the data pipeline into a single score that can be used and advertised to applications. This would help explain the interrelationship of the various stages of the BDC and data quality issues. This article implements a naive approach of considering all the scores equally, but the goal is to investigate other fusion techniques.

V. Rationale for Trust

This section serves and provides the motivation for introducing trust as a driver for data quality measurement and incorporating data quality into the big data model. Trust has been used previously as a measure of data quality. Keßler and De Groot [38] studied how the quality of geographic information can be estimated through the notion of trust as a proxy measure.

Trust is a unitless measure that can be used in composite metrics. A trust score can be used to represent the composite score of a chosen number of DQDs used to assess quality while representing a competitive score evaluating two (or more) sources. Moreover, trust based on previous events can help minimize the required processing time for real-time applications. Trust is expressed using the experience metric in the implementation of this article.

Experience (e):

Data quality is currently measured by evaluating over a period of data points and determining a quality score for an instance over this period. Real-time subscriptions to data streams are concerned with current data quality at the head of the stream. Assessing this quality over a period, up to the head of the stream can be expensive. Furthermore, this quality measure will age, requiring reevaluation.

This article presents an experience metric (e), based on a trust paradigm, which can be used to continuously assess quality at the head of a data stream with low overhead. Experience is modeled on the following properties of trust.

1) *Dynamic:* Trust can increase or decrease with new experiences (usage or interactions). This feature has been modeled through different techniques. For example, in PeerTrust [39], they use an iterative windowing approach which allows users to customize the overall trust score by varying past and present experiences of an actor. In this article, the defined generic experience metric provides an innovative way to allow a data stream/data provider to build trust over time. This is different from the current data quality evaluation strategies that, while considering historical data, return an instantaneous metric.
2) *Personalized:* This allows each data agent/consumer to customize its own trust metric by either assigning different weights to the metric or defining its own experience metric. This provides an innovative way to assess data quality based on the specifications of the data consumer.

VI. Proposed Framework

The framework is based on the properties of trust that can be improved over time to indicate good quality data whose quality threshold is acceptable for a certain use case. Trust itself is a continuous assessment process. Throughout this process, trust can be formed, improved, or lost. This article defines three trust stages for evaluating data quality. This was informed by previous research in the field [4]. Fig. 3 shows how these relate to the big data model.

1) *Initial trust:* This is trust that is derived without investigating the data itself but rather the context of the data. This includes investigating the source and equipment/sensors used, and metadata are also assessed. Metadata plays a key role in determining quality. Sensor

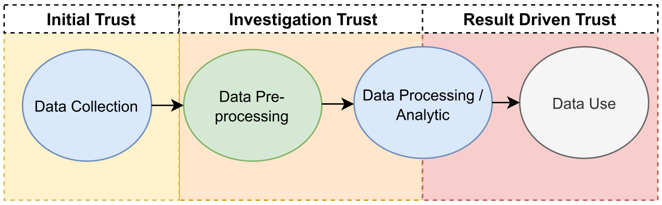


Fig. 3. Flow diagram showing the relationship between the big data model and trust formation stages.

types, manufacturer, deployment age, and calibration technique are features that affect data quality.

2) *Investigation trust:* Trust that is derived from the data itself. This can relate to the raw data stream or stages of preprocessing the data has undertaken. This, among other parameters, can encompass data completeness or accuracy. A component of this can consider comparative analysis, comparing the data stream values against a community of sensors or comparative data streams. Stages of preprocessing may involve data stream modeling or data fusion. Both modeling and fused data can be used and evaluated in comparative analysis for trust determination.

3) *Result-driven trust:* Data are used in systems modeling and decision-making. The result of these products can be monitored as effective decisions or accurate models. This monitored result can be used as a feedback mechanism throughout the stages of the big data model to build trust. This relates to the propagative property of trust [40]. The result-driven feedback is an important facet to build trust in the absence of a gold standard and in a shared data environment.

The initial trust would help us define a trust score that represents data quality during data collection. Investigative trust defines a trust score that represents data quality during data prepossessing and data analytics. Finally, result-driven trust defines a trust score that represents data quality during data analytics and other data use processes. This ensures that data quality is represented across the entire big data model. A proposed data quality assessment framework is shown in Fig. 4.

The framework defines three phases: 1) starting phase (SP); 2) investigation phase (IP); and 3) results phase (RP). These are explained further in Section VI-C. At each of the phases, a phase trust score is calculated; $\mathrm{SP}_n, \mathrm{IP}_n, \mathrm{RP}_n$ by combining parameters and weights. For example, SP (SP_n) is calculated with parameters $(a_1, a_2, \ldots, a_3)$ and weights $(w_1, w_2, \ldots, w_3)$. Table III presents a nonexhaustive list of parameters. This is completed for each of the phases with the respective parameters and weights. After determining a phase trust score, a use-case-specific threshold is defined. Such weights can be learned through feedback from the previous stage for a particular use case. This is used to model the effects of the phases on each other. The framework then combines these with the experience metric and outputs a single end-to-end trust metric that can be used to evaluate data quality.

TABLE III
PARAMETERS AT EACH PHASE

Phase	Parameters
Starting Phase	Timeliness, Documentation, Author of the data
Investigation Phase	Missing values, Number of parameters, Outliers
Results Phase	Successful model/analytics, Publications, Visualizations

A. Framework Phases

Three phases are defined that relate to the trust formation process. First, the framework will evaluate the context of the data by looking at aspects of the data; origin of data (reputation of the source), metadata. Second, the framework examines the data itself, assessing quality issues that exist in the data.

Finally, the framework assesses use-case-specific results and their applicability to data quality. A predictive model use case may compare real and predicted values and attribute trust based on this. At each phase, the framework defines parameters a_1 to a_n that define the dimensions of data quality used at each stage.

B. Determining Weights

The framework applies a set of linear weights to the attributes at each stage. The weights can be learned through the feedback process from the previous stage. This ensures that each use case can uniquely customize its own quality experience.

C. Formulation of the Trust Metric

The mathematical definition of (SP), (IP), and (RP) is illustrated further in Section VIII-B. The trust score of a data stream i is defined as

$$T_i = \mathrm{SP}_i + \mathrm{IP}_i + \mathrm{RP}_i + e \tag{1}$$

where e is a metric called experience defined below.

Experience (e): The proposed framework uses experience e to model the natural behavior of trust. This is motivated by work conducted by Gao *et al.* [41]. The experience score is driven by positive and negative experiences. Therefore, the metric e of a given data stream i is defined as

$$e_i = \frac{\vartheta_i + 1}{\vartheta_i + \delta_i + 2} \tag{2}$$

where ϑ_i is the count of positive experience toward data stream i, and δ_i is the count of negative experience toward data stream i.

Consider a data stream i, it is said to have positive e at time t if the trust score at time $t-1$ is greater than a threshold, else i shows a negative experience. At $t = 0$, $(\vartheta_i + \delta_i = 0)$, this means that data stream is new or has just started streaming. Therefore, experience is set to $e_i = 0.5$. This is informed by previous research in the field [42].

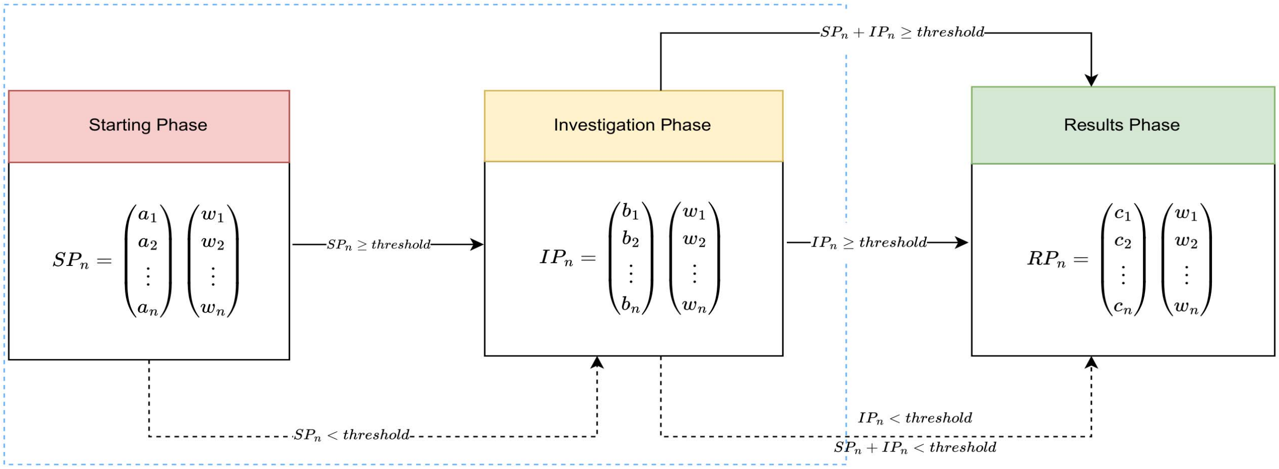

Starting Phase
$SP_n = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$
Investigation Phase
$IP_n = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$
Results Phase
$RP_n = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$
$SP_n \geq threshold$
$IP_n \geq threshold$
$SP_n + IP_n \geq threshold$
$SP_n < threshold$
$IP_n < threshold$
$SP_n + IP_n < threshold$

Fig. 4. Proposed framework that is based on trust formation stage.

VII. End-to-End Implementation

The proposed framework is tested via an end-to-end implementation of the data pipeline that is based on industry standard technologies and practices. The application development follows a microservice-based architecture that decomposes the application into a small set of complete self-contained services. This ensures that the framework can be seamlessly designed into any pipeline. This implementation aims to achieve the following goals.

1) Decouple the mechanics of providing or advertising a quality score from the methods of evaluating quality. This allows different, competing, or updated metrics to be used to calculate trust without affecting the scoring mechanism.
2) Describe the placement of the trust framework into big data pipelines.
3) Evaluate the feasibility of end-to-end data quality assessment in terms of computing resources.

The end-to-end data pipeline shown in Fig. 5 is composed of four system components: these are mapped from the four phases of the big data phases described in Section I. This section describes the role of each component and the technologies used in each.

A. Data Collection

This phase of the data pipeline is concerned with data producers or data sources. These could be live sensors streaming to the cloud, historical data coming from a data warehouse/database, or data coming from a third-party application programming interface (API). Both live stream and historical data sources (batch processing) were considered for testing and evaluation.

B. Data Preprocessing

Two operations take place in the preprocessing block; first, the calculation of the initial trust described in Section VI as part of the trust framework is applied. The system checks if the initial trust meets this stage's application data quality requirements. If satisfied, each data stream is tagged as good data and passed on to a Kafka node. Otherwise, that data can be tagged as usable data. In this case, the data can be improved by subsequent processing, and else, the data is discarded. Deciding what constitutes usable data is still an open research question.

Second, a producer–broker–consumer mechanism is set up to receive data from the data producers (sensors). This in turn will present the data to the next part of the data pipeline (consumer). In a typical IoT data-shared environment, it is expected that a single data source can share its data with one or several data consumers each with different data quality requirements. The need for parallel distributed processing where a single input data source can be processed independently by each application is supported by Kafka and Zookeeper.

Kafka, a distributed messaging system that is used for collecting and delivering high volumes of data with low latency [43]. Zookeeper is an orchestration server for Kafka. Fig. 6 shows the high-level architecture of Kafka. In the proposed system, each data consumer (spark application, described in Section VII-C) initializes a Kafka topic. This is the name of the source(s) from which it expects to receive data. A single application can subscribe to one or many Kafka topics. Each data source can only publish a single topic. Both Kafka and Zookeeper are run as microservices running on docker containers. The system can scale both horizontally and vertically.

C. Data Processing and Analytics

Data processing, real-time modeling, and data storage are performed by the data processing and analytics cluster. The second phase of trust calculation is conducted in this system component. To achieve the functionality of this subcomponent, the data needs to be processed in real-time as it is received. A spark streaming service is used to handle real-time data.

Spark streaming is part of the core spark API that allows for real-time processing of data from various sources including Kafka, Flume, and Amazon Kinesis. This processed data can then be saved to file systems, databases, live dashboards, or even pushed to a Kafka node as shown in Fig. 7.

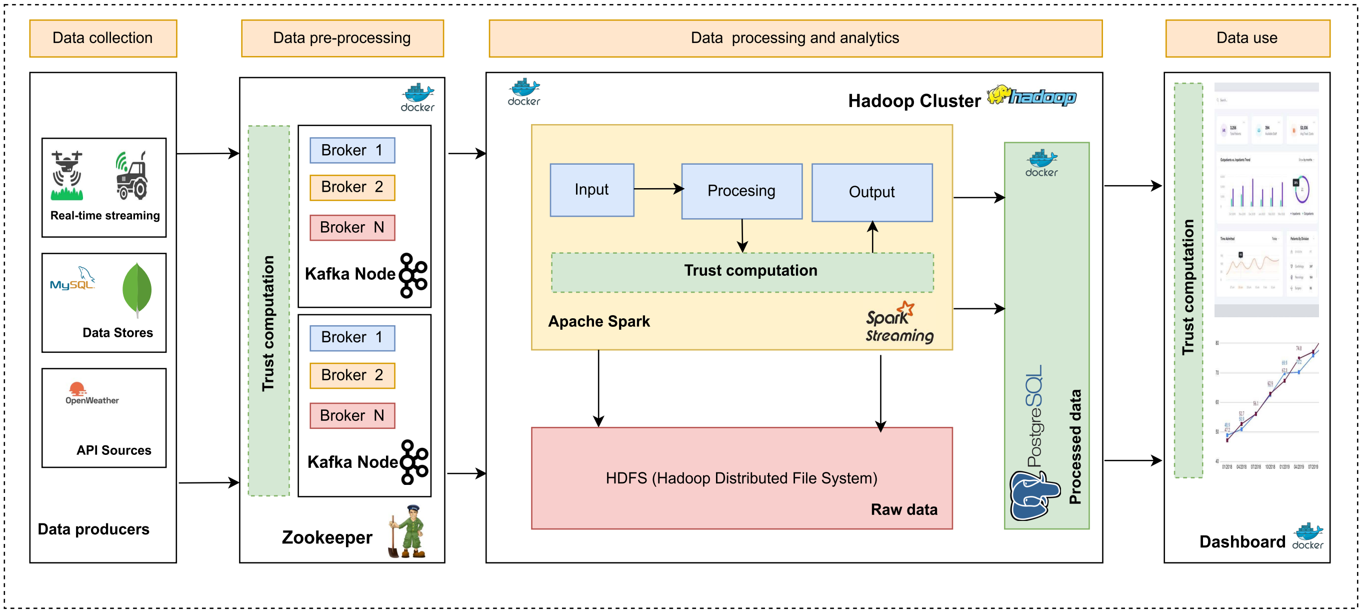


Fig. 5. End-to-end implementation of the data pipeline.

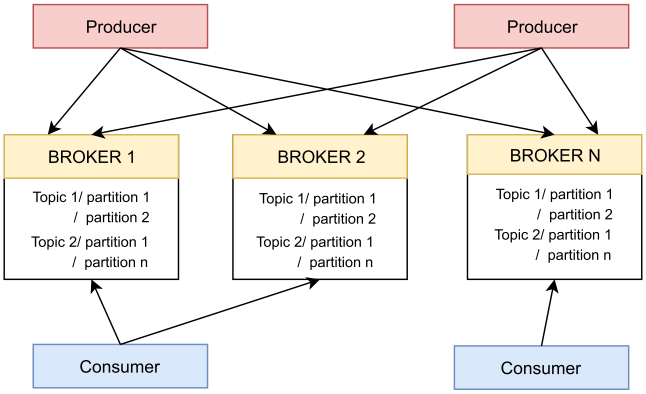


Fig. 6. Distributed Kafka architecture.

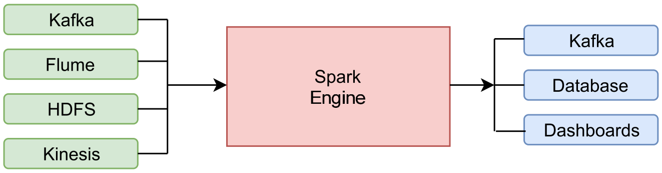


Fig. 7. Spark streaming application.

The setup consists of one master node with two workers. Each spark application is initialized with a list of topics to which it listens. Each topic corresponds to a data source name that is part of the IoT data-sharing ecosystem. Both historical and real-time data can be assessed using this implementation. Historical data can be assessed in batches defined by a period. Real-time data is assessed on a continuous basis windowed by a period or sample size.

The following operations take place using the calculated trust score.

1) The trust score from the previous phase is aggregated with that of the current phase.
2) If the trust score is lower than a threshold, operations can be performed here to improve the overall quality and a new trust score calculated. If the data does not meet the quality standards of the application, the current data source will be deemed unusable.

D. Data Use

The final component of the data pipeline considers data use. Data use operations such as data visualization and machine learning are typically applied in the BDC. RP trust is calculated based on the output of such operations. This requires feedback from the application. The feedback loop would help define the longitudinal relationship between the data quality properties at each stage of the big data model, and how these might affect or help improve the overall data quality of a data stream. For example, a prediction model application may return a prediction error based on the current data pipeline used.

VIII. Trust Framework by Example

To demonstrate the application of the proposed system, an example based on a dataset from an IoT deployment collected to measure air quality was used. The air quality dataset is comprised of two data streams: a gold reference sensor stream and an IoT sensor stream. Both are colocated and measure the same feature (carbon monoxide (CO) concentration). From each of these streams, a continuous trust metric is formed. The trust metric is built using three DQDs: 1) accuracy; 2) completeness; and 3) timeliness. The implementation of each of the metrics is defined in Section VIII-B. At any given time, a value of trust is given to the stream based on these DQDs and past trust scores (experience). The example is presented to, first, provide an implementation of trust and show the dynamics of the trust metric when evaluating a data stream, and second, to test and validate the trust mechanism. This is achieved in three stages.

1) First, the trust metric of the known gold standard data stream is compared against the trust metric for colocated IoT data stream. This test indicates how trust can be used as a metric over multiple DQDs to differentiate between sources of varying quality.
2) Second, trust is compared to known data quality metrics MAE and RMSE. The comparison will validate trust as an indicator of data quality. Should the defined trust metric show a strong correlation with the known data quality metrics over a range of data streams, we may conclude that the trust metric can indeed be a usable quality metric.
3) Finally, the resource costs of implementing trust as a continuous quality metric are evaluated.

The first evaluation is based on data from a CO sensor (referred to as the low-cost sensor) and a gold reference sensor. These are colocated and measure the same property. For each stream, DQDs at each stage are defined. These are used to calculate the trust score and subsequently experience for each stage. Finally, result scores from the initial and investigative stages are fused, resulting in a trust score for a stream. A naive approach using equal weighting linear fusion is applied in this example. Further investigation into fusion techniques will be conducted in future work.

A second evaluation is based on the gold reference stream. The goal is to have a standard and generic way of evaluating data quality frameworks that are based on model performance. Models are usually the final step of most IoT data processes, and the previous research has shown a correlation between data quality and model performance [44]. This can help establish a benchmark for all data quality frameworks across the IoT.

The stream is modeled to generate five streams. The first stream is the original stream (no noise added). The other five streams are generated by randomly introducing noise to the original stream in varying proportions of 10, 15, 20, 25, and 30%. The percentage is the overall size of the data stream.

To generate the noise, NumPy's random function was used [45]. This is based on the normal distribution and draws random samples. The size of the sample corresponds to the proportion of error, for example, 10%. The generated noise is then added to the original stream to create a new stream. For each stream, a trust metric is calculated. For each stream, a model is built and evaluated with two known metrics: 1) RMSE and 2) MAE. The framework is agnostic of the modeling technique. This example uses an autoregressive integrated moving average (ARIMA).

A relationship exists between data quality and model performance [44]. As the data quality degrades, so does model performance, keeping other factors constant. This relationship is used as a way to evaluate the trust metric. If this relationship exists between the calculated trust metric and data quality, it can be concluded that the trust metric is a valid metric for describing data quality. Fig. 8 summarizes the data and process flow of the example.

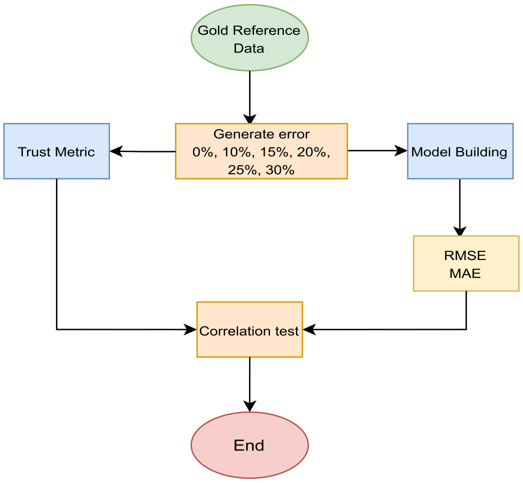


Fig. 8. Data and process flow of the example.

The final evaluation measures how the system resources are utilized as the data go through the data pipeline. Central processing unit (CPU) utilization of the system without the trust calculation and CPU utilization with the calculation of trust at each stage are compared. The delay introduced by trust calculation is also measured.

The results and evaluations reported in is study are based on the starting and investigation phases of the framework highlighted in Fig. 4. The mechanism for the RP is still under study. In the SP, one DQD is considered, timeliness, and two DQDs in the IP, completeness and accuracy. To differ from previous studies that use the same DQDs by taking into account only the current state of a data stream, our implementation uses a trust approach based on past and present experience. Karkouch *et al.* [46] reported that because of the instantaneous nature of these DQDs, at some point, they will be either unreliable or become insignificant. For example, if a sensor dirty fails (sensor node fails, but keeps up reporting readings that are erroneous) under any circumstances, then the accuracy dimension is rendered insignificant and unreliable unless it is enhanced with past experience. These metrics are defined in Section VI-C. This implementation considers uniform weights for each DQD. Effective weight determination is still an open research question.

A. Dataset Description

The study uses a publicly available dataset that was collected from an IoT deployment (see reference [47]). A multisensor device was colocated with a conventional air pollution analyzer. This was used to provide the true concentration values of the target pollutants at the measurement site. These values were thus used as a gold standard. This dataset is suitable for testing as the discrepancies in quality between the streams are known and can be used for system evaluation. This study uses data from the CO sensor (referred to as a low-cost sensor) and the gold reference sensor. Fig. 9 shows hourly concentration estimation of CO over a one-week window. The red line represents the true concentration value as measured by

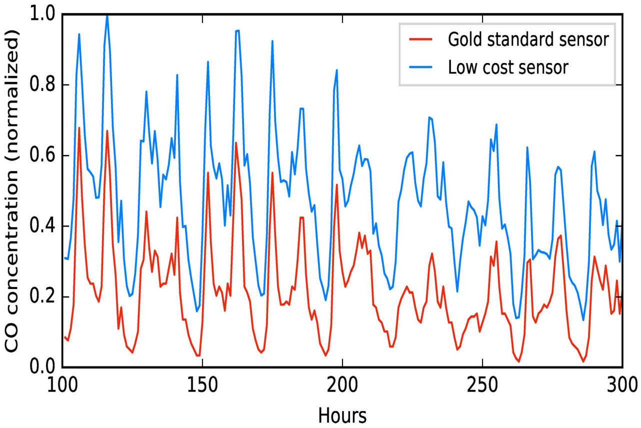


Fig. 9. Hourly concentration measures of CO for the gold standard and low-cost sensors.

the conventional analyzer colocated with the low-cost sensor whose values are shown by the blue line.

B. Mathematical Implementation

A previous study [2] examines if a trust-based framework can be used to evaluate the quality of a data stream without a gold reference. This study expands on this to consider the first two phases (SP and IP) of the framework. This work is different from the previous study in that it implements evaluation over the big data model, evaluating a fundamental component of the proposed framework.

The implementation and evaluation are conducted over two phases using: 1) timeliness and 2) accuracy and completeness DQDs. Following from (1), SP_i and IP_i are given by

$$\mathrm{SP}_i = (\text{Timeliness})\,(w_1) \tag{3}$$

$$\mathrm{IP}_i = \begin{pmatrix}\text{Accuracy}\\ \text{Completeness}\end{pmatrix}\begin{pmatrix}w_1\\ w_2\end{pmatrix}. \tag{4}$$

1) Accuracy: This is widely considered as meaning a correct and unambiguous correspondence with the real world [48]. Ballou and Pazer [49] defined accuracy as the recorded value being in conformity with the actual value. This work defines the metric for accuracy following the definition by Blake and Mangiameli [48]:

$$\text{Accuracy} := 1 - \frac{V_T}{N_A} \tag{5}$$

where V_T is the number of tuples in a relation having one or more incorrect values and N_A is the total number of tuples.

To determine V_T, a statistical technique based on median absolute deviations (MADs) is used. Absolute deviation from the median has been used for a long time to filter outliers [50]. The median is a measure of central tendency and is preferred to the mean as it is less sensitive to the presence of outliers which can have an outsized effect in IoT. The median is also a location estimator that has the highest breakdown point. The following formula as defined by Huber [51] was used to calculate MAD:

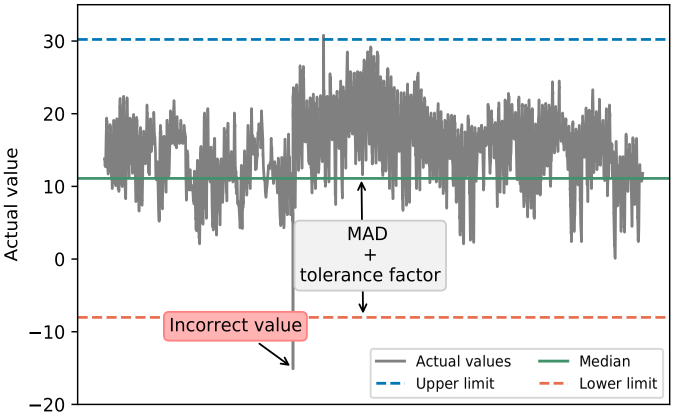


Fig. 10. Illustration of how MAD is implemented.

$$\text{MAD} = \alpha M_i(|x_i - M_j(x_j)|) \tag{6}$$

where x_j is the original observations, M_j is the median of the series, and α is the data normalization constant defined by [52]. It is defined as $\alpha = (1/(Q(0.75))$, where $Q(0.75)$ is the 0.75 quintile of that underlying distribution. The normalization step is important because otherwise MAD would estimate the scale up to a multiplicative constant [51] only.

Fig. 10 illustrates how the MAD was used to determine V_T. To determine the tolerance factor, Miller [53] suggested three values: 2, 2.5, and 3 standard deviations. The choice will determine the sensitivity of the metric. Since IoT data is noisy, the extreme value of 3 was used to ensure that noisy data are not tagged as outliers.

2) Completeness: The metric for completeness is given by [48] and is defined as follows: on the level of data values, a data value is incomplete (i.e., the metric value is zero) if and only if it is "NULL"; otherwise, it is complete (i.e., the metric value is one). All data values that represent missing or unknown values in a specific application scenario (e.g., blank spaces or "9/9/9999" as a date value) are represented by the data value "NULL." For a relation R, let T_R be the number of tuples in R that have at least one "NULL" value and let N_R be the total number of tuples in R. Then, the completeness of R is defined as follows:

$$\text{Completeness} := 1 - \frac{T_R}{N_R} = \frac{N_R - T_R}{N_R}. \tag{7}$$

3) Timeliness: The parameter for timelines of a data stream is affected by two components: currency, referring to the lag between when the data point was produced and when it was used or processed. The second, volatility, refers to how long the data point remains valid [54]. For some applications like accident avoidance in autonomous vehicles, this is very important. In others such as disease prediction in smart agriculture, the currency is less important. Unlike accuracy and completeness, timeliness is not determined directly from the data but rather by the context of the data. To this end, the metric for timelines is defined as

$$\text{Timeliness} = \left[\max\left(1 - \frac{\text{currency}}{\text{volatility}}\right), 0\right]. \tag{8}$$

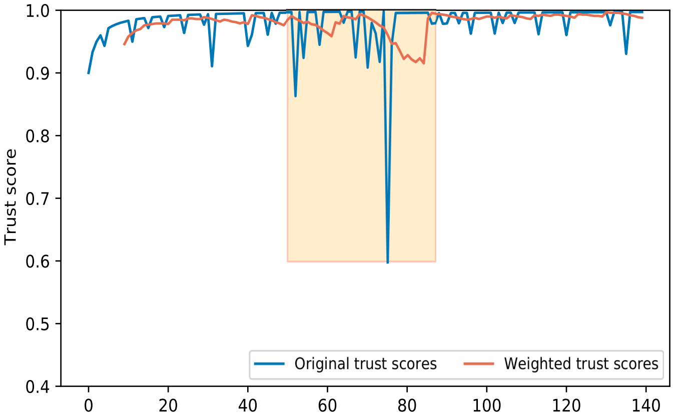


Fig. 11. Weighting trust scores using weighted moving average.

C. Weighting the Scores

Previous works [2] have two main drawbacks.

1) 1) Both previous and current experiences are weighted equally. The natural norm of trust assigns more weight to previous experiences when compared to current ones.
2) 2) The trust curve is biased by sudden changes in the data properties. The desired effect would be a gradual change for small changes in the data properties, with larger variations in trust occurring from larger data property changes.

To mitigate the above effects, a weighted moving average that assigns more weight to past experiences when compared to current experiences is applied. This also reduces the effects of sudden increases and decreases in the trust curve due to small changes. Fig. 11 compares the trust score before and after weighting. The highlighted area shows how the effects of weights mitigate the above challenges. This is important in data-shared IoT where data is noisy and sudden changes in the data do not always relate to poor quality data.

IX. Results and Evaluations

To evaluate the trust metric, an experiment was set up to compare the trust metric to known statistical measures. The ARIMA forecasting model was used. Although multiple factors dictate a model's performance, it is argued that the quality of the data that goes into the model is of most importance [44]. As the data quality changes, the assumption is that so does the model's overall performance. Using the original gold reference stream, a trust score was calculated. Also, an ARIMA model is built and tested for the same stream. The results of the trust score are then compared to the RMSE and MAE. The process is repeated for the other generated streams.

The correlation measures were based on the Pearson correlation coefficient. As it can be seen from both Table IV and Fig. 12, there is a strong negative correlation between the trust score and the RMSE and MAE. As the quality of the data degrades, the performance of the ARIMA model decreases. This is indicated by the increase in the values of RMSE and MAE. The same relationship exists between the trust score and data quality. As the quality of data degrades, so does the trust score. This shows that the proposed trust metric can act equivalently as known metrics like the RMSE. It is important to note that the trust metric is calculated without reference to a gold standard. This presents a significant opportunity to measure data quality in shared IoT where there is no gold reference to compare.

TABLE IV
Pearson Correlation Coefficients Between the Trust Score, RMSE, and MAE

	Mean trust score	RMSE	MAE
Mean trust score	1.0000	-0.9482	-0.9495
RMSE	-0.9482	1.0000	0.9831
MAE	-0.9495	0.9831	1.0000

The metric for RMSE and MAE are solely based on the accuracy of DQDs. It was, therefore, not possible to evaluate how completeness and timeliness would affect such metrics. Trust metric, however, is a multidimensional metric that incorporates several DQDs depending on the user's needs and application. This further illustrates the need for new ways to evaluate data quality. Future work will explore how to effectively compare the trust metric to other multidimensional metrics that support several DQDs.

Fig. 13 presents the data quality differences between the gold reference sensor and the low-quality sensor. The aim here was to differentiate between two known data quality streams: 1) the gold reference and 2) the low-cost sensor. The green highlight in the Fig. 13 shows how both the gold reference and low-quality sensor's trust decrease. During this period, the number of outliers increased by 2% for both the gold reference and low-quality sensors. The number of missing values, however, increased by 3% and 2% for gold reference and low-quality sensors, respectively. This accounts for the reduction in the trust scores for both sensors.

In the first red highlight, there are inconsistencies in the trust scores for the gold reference and low-quality sensors, with that of the low-quality sensor being lower. During this period, the data authors [47] reported that after the 30th week (starting march 2004), there was sensor drifting. This was later corrected by the calibration of the sensors. This sensor drift was detected by the trust mechanism and resulted in a lower trust score for this period. After calibration, the trust score returns to high values. The detection of sensor drift and its impact on data quality has previously shown to be difficult to measure.

In the last red highlight, there was a 13% increase in the number of missing values for the low-quality sensor. This explains the overall decrease in the trust score during this period. The trust metric, therefore, helps describe the quality of each data stream independently without relying on the other and shows the effects of different DQDs in a single metric. This type of comparison over multiple DQDs is not possible with existing techniques.

As part of the evaluation process, the impact of the trust computation framework on the data pipeline in terms of system resources (percentage CPU usage) and the delay introduced in the data pipeline due to trust calculation was monitored.

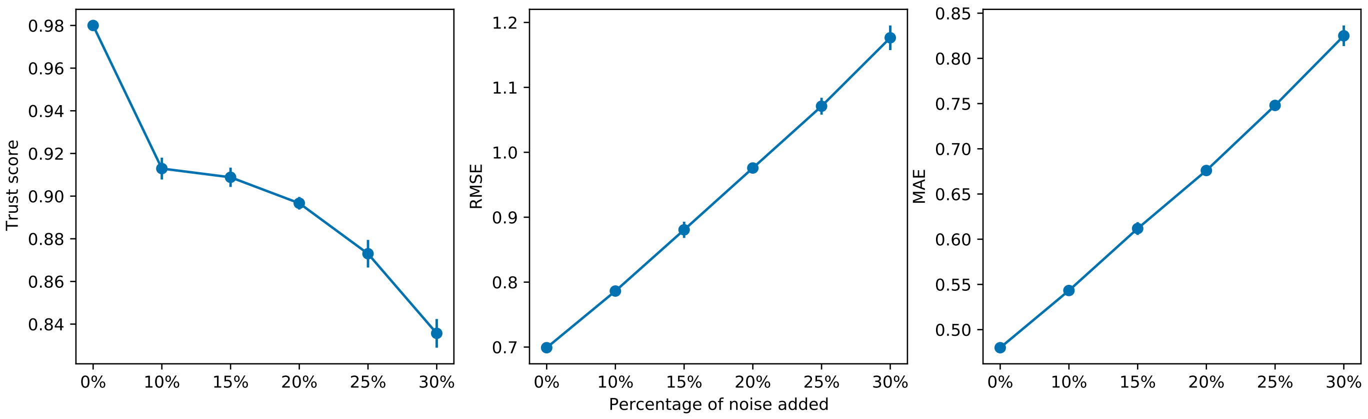


Fig. 12. Comparing how the trust score, RMSE, and MAE changes in data impurities.

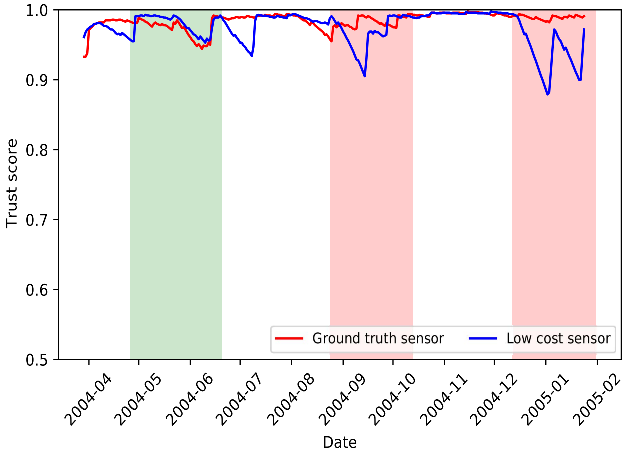


Fig. 13. Comparing trust scores for the gold reference sensor and low-cost sensor.

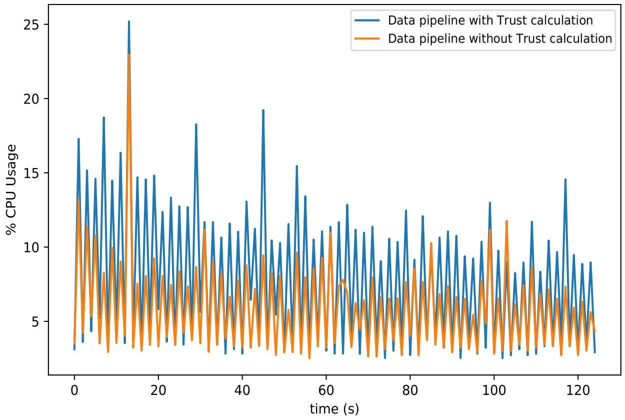


Fig. 14. Comparing data pipeline resource consumption.

TABLE V
SYSTEM RESOURCE UTILIZATION

No of worker nodes	Average compounded delay
2	10.6
3	7.5
4	4.2

Fig. 14 shows a higher percentage usage of compute resources for a data pipeline with trust computation when compared to the same job without trust computation. On average, there was a 7.8% increase in CPU usage for jobs with trust calculation. On average, there is a delay introduced into the data pipeline due to trust calculation. This is not an in-process delay, but rather compounded total delay from the start of the job to the end. Not with standing this the data is still delivered. The results reported here compare system performance as the number of nodes in the clusters increases (up to four worker nodes). Each of the worker's nodes has six cores and 4 GB. As shown in Table V, as the number of worker nodes increases, the delay decreases from 10.6 to 4.2 min. Since the delay is caused by an extra process (trust calculation), and spark is a distributed processing engine, increasing the number of nodes in the cluster would minimize or even eliminate such a delay. This demonstrates the scalability of the system.

X. APPLICATION: WHEN TO DISCONNECT FROM A DATA SOURCE

Data deriving from IoT can help foster innovations and save lives. Performing analytics on this data is, however, challenging due to the heterogeneity, complexity, and dynamic nature of IoT. Therefore, businesses and organizations have to maintain redundant data sources, seek third-party sources, or a few that can afford them, and install high-end sensors to mitigate such heterogeneity within the data.

The process of deciding which data source to engage and disengage is largely based on the quality of data from such a source. This is typically performed manually after the data has been processed. Data generation, collection, and delivery in IoT are automated and so should the process of selecting and maintaining a data source. Current data quality metrics are instantaneous, that is, a data source is evaluated based on its current state only, without previous context.

Given the dynamic, heterogeneity, and high volatility of IoT data, this would result in a high variance in network connection due to disconnecting from a data source each time there is a change in the data properties. This problem is exacerbated by low bandwidth and intermittent network disruptions that affect most IoT deployments.

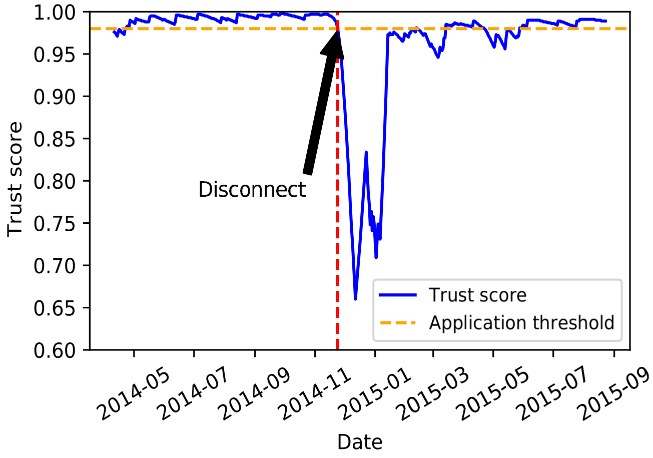


Fig. 15. Application with a higher threshold.

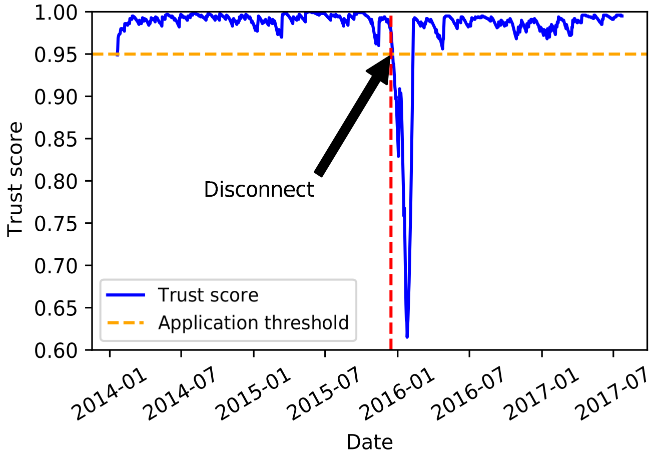


Fig. 16. Application with a lower threshold.

The trust metric, however, has been modeled to mitigate this problem. One of the novel elements of the trust score is its experience metric. This helps to incorporate the past and current context of a data source and also ensures that sudden changes in the data properties do not lead to sudden changes in the trust metric. This feature is further improved by the weighting strategy implemented in Section VIII-C.

To illustrate the application, data collected from weather stations between 2015 and 2019 in the United Kingdom was used. The data is produced at an interval of 15 min. Each weather station generates an average of 30 000 data points every year with over 100 weather stations. This includes air temperature, rainfall, relative humidity, and wind speed. Agricultural applications dynamically connect to and retrieve data from these stations.

In a data-shared IoT environment, applications should be able to dynamically select a data source, a weather station, for example, based on its data quality needs. Figs. 15 and 16 show two applications each with a different trust score threshold (0.98 and 0.95), each connected to a different weather station. Each application would advertise its threshold. In each case, the application would then automatically disengage from the data source when its trust score falls below the application's threshold and would automatically be connected to another data source of sufficient quality. While the thresholds determined in the figures are somewhat arbitrary, the method can be used to allow different applications to prescribe different levels of quality for their given use case. Each application and use case can set its own threshold. This can then be used to disengage from a data source.

This kind of automation can be used in other applications, for example:

1) 1) Automate data source repairs and maintenance. Continued disengagement from a data source could mean it is faulty.
2) 2) Gradual decrease in data quality could be associated with sensor drift. Therefore, the sensor can be automatically reset back to where its known data quality was good.

XI. Conclusion and Future Work

This article has described an end-to-end implementation of a trust framework that can be used to estimate the quality of data. The implementation was evaluated using data collected from a real-world experiment.

The implementation is based on industry-standard data pipelines with Kafka as distributed messaging service and Apache Spark for real-time data processing. This is unique as it integrates with all the stages of the big data model. To the best of our knowledge, it is the *first* end-to-end data quality assessment framework implementation. This implementation enables one to estimate data quality in cases where there is no gold standard to compare. The other advantage is that one would be able to represent data quality in a general manner throughout the big data model.

To evaluate the system, the article compares the trust score to RMSE and MAE. Using Pearson's correlation coefficient, the results and evaluations showed that the trust metric is a good metric for data quality assessment in cases where there is no gold standard, which is the case in data-shared IoT. Although the results have shown a slight increase in system resources in terms of CPU and execution time, this can be mitigated with distributed processing.

The article listed several challenges related to data quality assessment, both structural and method-related challenges. We have explored how some of these can be solved by the implementation described using real-world datasets. However, some challenges remain: for example, we equally combine the trust scores for the different phases. There is a need for a fusion algorithm that accounts for the contribution of each stage. Also, the mechanism for feedback and how this propagates from the RP is not yet fully formed. These and other challenges highlighted above form part of our future work.

References

[1] G. A. Lakshen, S. Vranes, and V. Janev, "Big data and quality: A literature review," in *Proc. 24th Telecommun. Forum (TELFOR)*, Nov. 2016, pp. 1–4.

[2] J. Byabazaire, G. O'Hare, and D. Delaney, "Using trust as a measure to derive data quality in data shared IoT deployments," in *Proc. 29th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2020, pp. 1–9.

[3] F. Sidi, P. H. S. Panahy, L. S. Affendey, M. A. Jabar, H. Ibrahim, and A. Mustapha, "Data quality: A survey of data quality dimensions," in *Proc. Int. Conf. Inf. Retr. Knowl. Manage.*, Mar. 2012, pp. 300–304.

[4] I. M. Faniel and T. E. Jacobsen, "Reusing scientific data: How earthquake engineering researchers assess the reusability of Colleagues' data," *Comput. Supported Cooperat. Work*, vol. 19, nos. 3–4, pp. 355–375, Aug. 2010.

[5] A. Yoon, "Data reusers' trust development," *J. Assoc. Inf. Sci. Technol.*, vol. 68, no. 4, pp. 946–956, Apr. 2017.

[6] Y. Kim and A. Yoon, "Scientists' data reuse behaviors: A multilevel analysis," *J. Assoc. Inf. Sci. Technol.*, vol. 68, no. 12, pp. 2709–2719, Dec. 2017.

[7] I. Taleb, M. A. Serhani, C. Bouhaddioui, and R. Dssouli, "Big data quality framework: A holistic approach to continuous quality management," *J. Big Data*, vol. 8, no. 1, pp. 1–41, Dec. 2021.

[8] N. U. Okafor and D. Delaney, "Considerations for system design in IoT-based autonomous ecological sensing," *Proc. Comput. Sci.*, vol. 155, pp. 258–267, Jan. 2019.

[9] P. Z. Yeh and C. A. Puri, "An efficient and robust approach for discovering data quality rules," in *Proc. 22nd IEEE Int. Conf. Tools Artif. Intell.*, Oct. 2010, pp. 248–255.

[10] F. Chiang and R. J. Miller, "Discovering data quality rules," *Proc. VLDB Endowment*, vol. 1, no. 1, pp. 1166–1177, Aug. 2008.

[11] W. Fan, "Data quality: Theory and practice," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Berlin, Germany: Springer, 2012.

[12] I. Taleb, M. A. Serhani, and R. Dssouli, "Big data quality: A survey," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, Jul. 2018, pp. 166–173.

[13] S. Kandel *et al.*, "Research directions in data wrangling: Visualizations and transformations for usable and credible data," *Inf. Vis.*, vol. 10, no. 4, pp. 271–288, 2011.

[14] M. Chen, M. Song, J. Han, and E. Haihong, "Survey on data quality," in *Proc. World Congr. Inf. Commun. Technol., (WICT)*, 2012, pp. 1009–1013.

[15] M. Heravizadeh, J. Mendling, and M. Rosemann, "Dimensions of business processes quality (QoBP)," in *Business Process Management Workshops* (Lecture Notes in Business Information Processing). Berlin, Germany: Springer, 2009.

[16] H. Baqa, N. B. Truong, N. Crespi, G. M. Lee, and F. Le Gall, "Quality of information as an indicator of trust in the Internet of Things," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Communications/ 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 204–211.

[17] S. Juddoo, "Overview of data quality challenges in the context of big data," in *Proc. Int. Conf. Comput., Commun. Secur. (ICCCS)*, Dec. 2015, pp. 1–9.

[18] Y. W. Lee, D. M. Strong, B. K. Kahn, and R. Y. Wang, "AIMQ: A methodology for information quality assessment," *Inf. Manage.*, vol. 40, no. 2, pp. 133–146, Dec. 2002.

[19] Z. Malik and A. Bouguettaya, "Reputation bootstrapping for trust establishment among web services," *IEEE Internet Comput.*, vol. 13, no. 1, pp. 40–47, Jan. 2009.

[20] E. Chang, T. Dillon, and K. F. Hussain, *Trust and Reputation for Service-Oriented Environments: Technologies for Building Business Intelligence and Consumer Confidence*. Chichester, U.K.: Wiley, 2006.

[21] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decis. Support Syst.*, vol. 43, no. 2, pp. 618–644, Mar. 2007.

[22] J. O'Donovan and B. Smyth, "Trust in recommender systems," in *Proc. Int. Conf. Intell. User Interfaces*, 2005, pp. 167–174.

[23] J. Byabazaire, G. O'Hare, and D. Delaney, "Data quality and trust: Review of challenges and opportunities for data sharing in IoT," *Electronics*, vol. 9, no. 12, p. 2083, Dec. 2020.

[24] L. Zhang, D. Jeong, and S. Lee, "Data quality management in the Internet of Things," *Sensors*, vol. 21, no. 17, p. 5834, Aug. 2021.

[25] M. Chen, M. Song, J. Han, and E. Haihong, "Survey on data quality," in *Proc. World Congr. Inf. Commun. Technol.*, 2012, pp. 1009–1013, doi: 10.1109/WICT.2012.6409222.

[26] K. Fizza, P. P. Jayaraman, A. Banerjee, D. Georgakopoulos, and R. Ranjan, "Evaluating sensor data quality in Internet of Things smart agriculture applications," *IEEE Micro*, vol. 42, no. 1, pp. 51–60, Jan. 2022.

[27] R. Y. Wang, "A product perspective on total data quality management," *Commun. ACM*, vol. 41, no. 2, pp. 58–65, 1998.

[28] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," *Commun. ACM*, vol. 45, no. 4, pp. 211–218, 2002.

[29] O. Azeroual, J. Schöpfel, D. Ivanovic, and A. Nikiforova, "Combining data lake and data wrangling for ensuring data quality in CRIS," in *Proc. 15th Int. Conf. Current Res. Inf. Syst. (CRIS)*, 2022.

[30] M. Abdallah, A. Hammad, and W. AlZyadat, "Towards a data collection quality model for big data applications," in *Proc. Int. Conf. Bus. Inf. Syst.*, 2022, pp. 103–108.

[31] C. Batini and A. Rula, "From data quality to big data quality: A data integration scenario," in *Proc. SEBD*, 2021, pp. 36–47.

[32] N. Gaviria *et al.*, "Data quality estimation in a smart City's air quality monitoring IoT application," in *Proc. 2nd Sustain. Cities Latin Amer. Conf. (SCLA)*, Aug. 2021, pp. 1–6.

[33] R. Alrae, Q. Nasir, and M. A. Talib, "Developing house of information quality framework for IoT systems," *Int. J. Syst. Assurance Eng. Manage.*, vol. 11, no. 6, pp. 1294–1313, Dec. 2020.

[34] F.-K. Tsai, C.-C. Chen, T.-F. Chen, and T.-J. Lin, "Sensor abnormal detection and recovery using machine learning for IoT sensing systems," in *Proc. IEEE 6th Int. Conf. Ind. Eng. Appl. (ICIEA)*, Apr. 2019, pp. 501–505.

[35] E. Vilenski, P. Bak, and J. D. Rosenblatt, "Multivariate anomaly detection for ensuring data quality of dendrometer sensor networks," *Comput. Electron. Agricult.*, vol. 162, pp. 412–421, Jul. 2019.

[36] N. Javed and T. Wolf, "Automated sensor verification using outlier detection in the Internet of Things," in *Proc. 32nd Int. Conf. Distrib. Comput. Syst. Workshops*, Jun. 2012, pp. 291–296.

[37] D.-Y. Kim, Y.-S. Jeong, and S. Kim, "Data-filtering system to avoid total data distortion in IoT networking," *Symmetry*, vol. 9, no. 1, p. 16, Jan. 2017.

[38] C. Keßler and R. T. A. de Groot, "Trust as a proxy measure for the quality of volunteered geographic information in the case of openstreetmap," in *Geographic Information Science at the Heart of Europe* (Lecture Notes in Geoinformation and Cartography). Cham, Switzerland: Springer, 2013.

[39] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, Jul. 2004.

[40] W. Sherchan, S. Nepal, and C. Paris, "A survey of trust in social networks," *ACM Comput. Surveys*, vol. 45, no. 4, pp. 1–33, Aug. 2013.

[41] Y. Gao, X. Li, J. Li, Y. Gao, and P. S. Yu, "Info-Trust: A multi-criteria and adaptive trustworthiness calculation mechanism for information sources," *IEEE Access*, vol. 7, pp. 13999–14012, 2019.

[42] E. J. Friedman and P. Resnick, "The social cost of cheap pseudonyms," *J. Econ. Manage. Strategy*, vol. 10, no. 2, pp. 173–199, Jun. 2001.

[43] J. Kreps, N. Narkhede, and J. Rao, "Kafka: A distributed messaging system for log processing," in *Proc. ACM SIGMOD Workshop Netw. Meets Databases*, 2011, pp. 1–7.

[44] V. Sessions and M. Valtorta, "The effects of data quality on machine learning algorithms," in *Proc. Int. Conf. Inf. Qual., (ICIQ)*, 2006, pp. 485–498.

[45] The NumPy Community. *Numpy Random Normal Function*. Accessed: Oct. 27, 2021. [Online]. Available: http://numpy.random.normal

[46] A. Karkouch, H. Mousannif, H. A. Moatassime, and T. Noel, "Data quality in Internet of Things: A state-of-the-art survey," *J. Netw. Comput. Appl.*, vol. 73, pp. 57–81, Sep. 2016.

[47] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sens. Actuators B, Chem.*, vol. 129, no. 2, pp. 750–757, Feb. 2008.

[48] R. Blake and P. Mangiameli, "The effects and interactions of data quality and problem complexity on classification," *J. Data Inf. Qual.*, vol. 2, no. 2, pp. 1–28, Feb. 2011.

[49] P. D. Ballou and L. H. Pazer, "Impact of inspector fallibility on the inspection policy in serial production systems," *Manage. Sci.*, vol. 28, no. 4, pp. 387–399, 1982.

[50] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *J. Exp. Social Psychol.*, vol. 49, no. 4, pp. 764–766, Jul. 2013.

[51] P. J. Huber, *Robust Statistical Procedures*. Philadelphia, PA, USA: SIAM, 1996.

[52] P. J. Rousseeuw and C. Croux, "Alternatives to the median absolute deviation," *J. Amer. Stat. Assoc.*, vol. 88, no. 424, pp. 1273–1283, Dec. 1993.

[53] J. Miller, "Short report: Reaction time analysis with outlier exclusion: Bias varies with sample size," *Quart. J. Exp. Psychol. Sect. A*, vol. 43, no. 4, pp. 907–912, Nov. 1991.

[54] D. Ballou, R. Wang, H. Pazer, and G. K. Tayi, "Modeling information manufacturing systems to determine information product quality," *Manage. Sci.*, vol. 44, no. 4, pp. 462–484, Apr. 1998.

John Byabazaire received the B.Sc. degree in computer science from Gulu University, Gulu, Uganda, in 2013, and the M.Sc. degree (by research) from Waterford Institute of Technology, Waterford, Ireland, in 2018. He is currently pursuing the Ph.D. degree with the School of Computer Science, University College Dublin, Dublin, Ireland.

Alongside his current Ph.D. study, he continues to research e-learning for low-bandwidth environments, software-defined networking, network function virtualization, and remote sensing, the Internet of Things (IoT), and fog analytics. His research interests include IoT systems for the data collection on farms. This is part of the CONSUS Project (https://www.ucd.ie/consus/).

Gregory M.P. O'Hare (Member, IEEE) is a Professor of Artificial Intelligence and the Head of the School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland, and a Visiting Professor at UCD, Dublin. He has over 500 refereed publications of which over 100 are in high-impact journals. He has edited ten books and has a cumulative career research grant income of €82 million. He is an established Principal Investigator with Science Foundation Ireland, has been one of the founders of the CLARITY Centre (now INSIGHT), and is an SFI Principal Investigator on the Project CONSUS (https://www.ucd.ie/consus/), an SFI industry-funded collaboration with Origin Enterprises focused on precision agriculture. His research interests include artificial intelligence and multiagent systems (MAS), ubiquitous computing, and wireless sensor networks.

Declan T. Delaney received the Ph.D. degree in network analysis and design for IoT from the School of Computer Science, UCD, Dublin, Ireland, in 2015.

He is currently an Assistant Professor at the School of Electrical and Electronic Engineering, UCD. Previously, he worked at LMI Ericsson, Dublin, and collaborations with SMEs in H2020 funding proposals. He maintains strong links with industry partners. He is an SFI Funded Investigator on the Project CONSUS (https://www.ucd.ie/consus/), an SFI industry-funded collaboration focused on precision agriculture, and a Principal Investigator for the SmartBOG Project (www.smartbog.com). His research interests include network data analytics for adaptable programmable networks and infrastructure and data assurance for the IoT and sensor systems.