AI Quality Engineering for Machine Learning Based IoT Data Processing

Chapter *in* Communications in Computer and Information Science · December 2022

DOI: 10.1007/978-3-031-21637-4_4

2 authors:

Shelernaz Azimi
Free University of Bozen-Bolzano
16 PUBLICATIONS **118** CITATIONS
SEE PROFILE

Claus Pahl
Free University of Bozen-Bolzano
431 PUBLICATIONS **8,947** CITATIONS
SEE PROFILE

AI Quality Engineering for Machine Learning based IoT Data Processing

Shelernaz Azimi, Claus Pahl

Free University of Bozen-Bolzano, Bolzano, Italy
{fname.sname}@unibz.it

Abstract. Raw source data can be made accessible in the form of processable information through Data-as-a-Service (DaaS) architectures. Machine learning is one possible way that allows to produce meaningful information and knowledge based on this raw source data. Thus, quality is a major concern that applies to raw data as well as to information provided by ML-generated models. Quality management is a major concern of AI Engineering – an attempt to systematically produce quality AI solutions. As the core of our solution, we define a conceptual framework that links input data quality and the machine learned data service quality, specifically inferring raw data problems as root causes from observed data service deficiency symptoms. This will allow to identify the hidden origins of quality problems that might be observed by users of DaaS offerings. We analyse the quality framework using a real-world case study from an edge cloud and Internet-of-Things-based traffic application. We identify quality assessment techniques for symptom and cause analysis.

Keywords: AI Engineering, Data-as-a-Service, DaaS, Machine Learning, Continuous Quality Management, Edge Cloud, Internet-of-Things, Traffic System.

1 INTRODUCTION

Raw source data can be made accessible for the consumer of the service in the form of processable information through Data-as-a-Service (DaaS) architectures. A problem here is that quality concerns observed by the consumer of the service are caused by quality problems related to the raw source data or its processing – both of which are hidden from the consumer and not directly observable.

We propose an AI Engineering solution, more specifically a continuous data and machine learning (ML) model quality management in order to deal with an ongoing process of continuously monitoring and improving the quality of data and derived ML models. In particular, in contexts dominated by high volume, velocity and veracity of data (generally referred to as big data), which includes out Data-as-a-Service (DaaS) setting, this continuous quality management process is crucial. Data processing using ML techniques is an integral part of obtaining value out of the raw data, but require a dedicated quality management approach. We build on data quality models, extending the data quality concept to the ML model level. Also, we need to close the loop by mapping quality problems (the symptoms) at ML level back to their origins, i.e., aiming at a root cause analysis solution.

We present a layered data architecture for both data and ML function layers and enhance this by a root cause analysis based on a closed loop between two layers. We determine quality assessment mechanisms for symptom and cause analysis in different dimensions, including situational analysis and timeseries, determination outcome, object, type and techniques. Our approach is suited to situations where raw data quality might not be directly observable or assessable, thus a new way of inferring quality is needed. In order to validate our proposed solution, we use a case study. The context is a public data service (DaaS) application, specifically at a regional level (a regional IT and Data service provider). The application is traffic management, which is based on traffic and weather data collected locally in an edge cloud and IoT setting. The novelty of this paper, which extends [5], lies in the layering of data and model quality based on dedicated ML function types,

In Section 2, we introduce the principles of Explainable AI and AI Engineering, befre outlining the DaaS quality management framework. The details of the solution are presented in Section 4. In Section 5, we discuss the IoT traffic use case. Related work is discussed in Section 6 and we conclude in Section 7.

2 Explainable AI and AI Engineering

Explainable AI is the context of this work. Explainability is the extent to which the internal mechanics of a machine or deep learning system can be explained in human terms. Interpretability is about being able to identity the mechanics without necessarily knowing why. Explainability is being able to explain what is happening. Our ultimate objective is to automate a root cause analysis that aims to 'explain' the reasons for quality deficiencies or defects [20] in the ML model. This explains ML quality in terms of data quality [27].

Applied to our Internet-of-Things (IoT) setting, this means that for instance accuracy problems with traffic or weather prediction models or often cause by either unsuitable ML model construction or by data quality problems of data that is processed by the ML models. Here, we are specifically interested in understanding the impact of IoT data quality concerns. This in concrete terms meaning to understand if sensor failures cause incorrect readings or if network outages cause the data to be incomplete.

Overall, the aim is to move towards an explainability or interpretability of ML model failures/deficiencies as an a-posteriori measure for detection and correction [48]. Pre-construction data validation is an advisable step prior to model construction. In contrast to works in this context, we aim to identify missing values/default replacements as the root cause of prediction deficiencies (such as accuracy) as a remedial action. Some problems will still go undetected in an a pre-construction approach. Our approach (an a-posteriori analysis) can be adjusted to the presence of a-priori validation of data. Our approach also allows a black-box mode, if the construction itself is not visible/observable.

AI Engineering [7] is working towards a systematic construction of for instance ML models in order to achieve and maintain quality. Our root cause analysis can also been seen as an endeavour to continuously improve quality.

3 Continuous AI Data and Model Quality Management

3.1 Quality Principles

A continuous quality management for *data services* is a continuous process of data quality actions, namely prevention, detection and correction. The prevention of problems is, however, not always achievable. Consequently, we focus in this paper on the detection and correction of quality problems.

We target the quality of information models that are generated from data using ML techniques. Data is an asset in the IoT domain as a source for creating valuable information and knowledge. Quality is a critical requirement for the data consumer (e.g. IoT pervasive services and their users) in this context Data quality refers to how well data meets the requirements of its users. Sample aspects are accuracy, timeliness or completeness [49]. Based on this broader conceptualization and 159 defined dimensions, four main categories have been identified: Intrinsic, Contextual, Representational, Accessibility.

Quality frameworks for data and information have been investigated for a while. [33, 4, 3]. There is also an accepted classification of (big) data aspects that helps us in organising the quality concerns [41, 32], often referred to as the *4V model*:

- volume (scale, size),
- velocity (change rate/streaming/real-time),
- variety (form/format) and
- veracity (uncertainty, accuracy, applicability).

Note, that sometimes value is added as a fifth concern, but we focus on the technical aspects here.

3.2 IoT Use Case

Our chosen IoT application domain shows all of these four main characteristics. In the *Edge Cloud* and *Internet-of-Things* (IoT), so-called things (like sensors and actuators) produce and consume data [36], processed in a edge cloud, in order to provide DaaS services.

In case the underlying data is inaccurate, any extracted information and also derived actions based on it are likely to be unreliable [26]. Thus, *data quality* concerns arise. Furthermore, the edge cloud environment in which the data collection occurs is often rapidly changing in terms of architecture and data characteristics. In order to focus our investigation, we make the following assumptions: (i) all data is numerical (i.e., text or multimedia data and corresponding quality concerns regarding formatting and syntax are not considered) and (ii) data can be stateful or stateless. Thus, IoT is a 4V big data context with specific data types, making our results transferable to similar technical environments.

The following two problems shall be addressed in our setting:

- *(1) Quality Value Analysis:* is based on quality goals and thresholds. Goals are defined in terms of quality dimensions such as accuracy or completeness. The reaching of goals is determined using predefined thresholds.

– *(2) Problem Cause Analysis and Prediction:* rely on pattern and anomaly detection to identify DaaS information model quality problems and map them the data layer, possibly including time series such as quality graphs over time (at DaaS and source data level).

The problem is if a problem source at the data layer can be identified or predicted based on an analysis of the DaaS layer, i.e., whether a root cause analysis is possible.

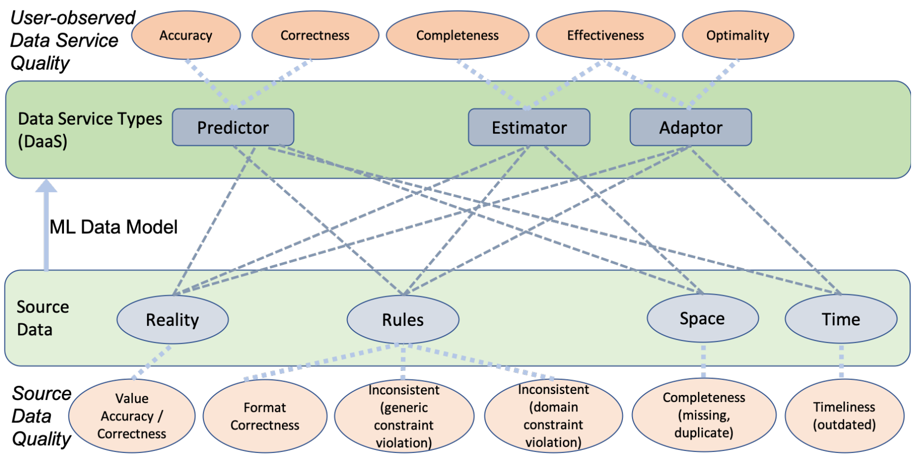

User-observed Data Service Quality
Accuracy
Correctness
Completeness
Effectiveness
Optimality
Data Service Types (DaaS)
Predictor
Estimator
Adaptor
ML Data Model
Source Data
Reality
Rules
Space
Time
Source Data Quality
Value Accuracy / Correctness
Format Correctness
Inconsistent (generic constraint violation)
Inconsistent (domain constraint violation)
Completeness (missing, duplicate)
Timeliness (outdated)

Fig. 1: A Layered DaaS Quality Management Architecture (see [5]).

4 DaaS Quality Assessment & Problem Cause Analysis

A number of quality concerns such as accuracy or completeness have been identified in an empirical study [14] for ML data models. Our objective is to associate these concerns systematically to different root causes.

The differentiation can help to better identify IoT-level root causes for observed problems:

– *Problems with IoT input data.* Assume a data table `'TrafficCount(Location, Date/Hour, Direction1, Direction2)'`. Two types of data quality problems are: (i) missing values (e.g., for one direction), which could result from a single sensor failure, and (ii) missing record (e.g., all data for a whole hour or from one location), which could result from communication failure.
– *Problems with ML data model training.* Here unsuitable training sets (e.g., incomplete) could have been used in the construction.

4.1 Quality Layers

The starting point that forms the basis of the data quality architecture is the raw or source data layer, see Fig. 1. We can distinguish context-independent data qualities (complete, missing, duplicate, correct/accurate value, correct format, timely/outdated, inconsistent/violation of generic constraint) and context-dependent data quality (violation of domain constraints). Raw (source) data is consumed by to produce machine learning models or functions. Data concerns can be grouped:

– Reality: value accuracy and correctness - here quality concerns are gathered that can be verified against the reality,
– Rules: format correctness, consistency (generic constraints), consistency (domain constraints) - these encompass criteria usually applied to data in databases from structural/format aspects to domain-specific constraints,
– Space: completeness - this refers to completeness as a 'spacial' constraint,
– Time: timeliness - this is an attribute that indicates the timely availability of data in the system in relation to its creation time.

In order to better understand the processing purpose, we categorise these functions into DaaS function types: *predictor*, *estimator* (or calculator) and *adaptor*. For these functions, we define an information quality model. In the literature, estimation and prediction as function types are often not clearly enough distinguished. Here, we characterise them as follows [13]:

– An estimator is a rule for computing a quantity from a sample that is to be used to estimate a model parameter. An example is the average time taken for some task (for given sample). There is Low variance good (avoid noise, but can oversimplify/underfit) because of the smaller confidence interval. High variance would have a lot of detail, but miss the main feature. Guessing/estimating the main feature is desired. In comparison, there is less data, obtained in a structured, planned way (but could still be inaccurate, cf/ hospital studies). Furthermore, somewhat larger resources aids this. Generally, the main feature is fine, but any type of noise can be filtered out and simplified (but continuous and smooth). With comparably less resources, data accuracy and completeness are more of a problem. Regression is a good technique for this
– Prediction is to predict a new observation (i.e. one that is not in the sample). An example is the next predicted time another person takes to do something The point estimate (average) is probably the same, but the confidence interval (e.g. 95%) is generally wider: here there is variance on the sample and also for obtaining the new observation (value of the random variable y in one trial of an experiment, and not just to construct a confidence interval in the same sense that we have done before). Models can be discrete and jagged (rather than continuous and smooth). Low bias is good (but can do overfitting), i.e. have high precision to reduce confidence interval and be more definitive, i.e., getting a better result for the one new observation. In comparison, there is more data, but less systematically obtained (cf. image collection). Often, there is a vast amount of data, but mixed and not homogeneous. Data uncertainty (inconsistency) is more of a problem. Neural networks are good for this.

To these two widely used concepts, we add the *adaptor*:

- Adaption refers to the conversion of a prediction or estimation into an action that results from these. The difference to the previous two is that the quality concern is effectiveness of the action, not the predicted or estimated value. What is measured is how well the action remedies any undesired predicted or estimated value. Optimising effectiveness rather than accuracy is the main quality objective of this function. An example is a controller that manages the workload of a machine. Machine configuration parameters can by (if necessary, dynamically) reconfigured through the calculated action in order to optimise the productivity.

The evaluation of our use case will shows that we can relate DaaS function quality to DaaS function types and techniques, see Fig. 1: Predictors are concerned with accuracy (regression) and correctness (classification). Estimator are concerned with effectiveness (clustering) and completeness (clustering). Adaptors are concerned with effectiveness (classification) and optimality (regression). Input for function quality includes thus the following aspects:

- structural model quality: accuracy, correctness, completeness, effectivess, optimality and
- function-specific quality: accuracy/correctness (for the predictor), complete/effective (for the estimator), effectiveness/optimality (for the adaptor).

Furthermore ethical model/function quality includes fairness, privacy-preserving. Note, that is essential here to assess the quality of the function provided by the ML models, which emerge in different types, such as predictors, estimators or adaptors. Often multiple goals involved. For example, primary goal effective (performance threshold) with secondary goal optimality (energy/resources) to maintain the threshold.

4.2 MAPE-K Service Quality Loop

The quality management for our DaaS architecture includes a feedback loop based on the MAPE-K adaptation pattern (with its Knowledge-based phases Monitor, Analyse, Plan and Execute) to control data and information quality, see Fig. 2. At the core is a mapping of DaaS model quality to source data quality based Fig. 1 presented earlier.

We start with accuracy, which is often considered the most important quality concern. High precision relates to a low false positive (FP) rate

$$TP/(TP+FP)$$

i.e., correctly identified over incorrectly identified. High recall relates to a low false negative (FN) rate

$$TP/(TP+FN)$$

i.e., correctly identified over not identified correct cases. High precision means that a DaaS function returns substantially more relevant results than irrelevant ones, while high recall means that it returns most of the relevant results. Correctness plays a significant role in this function.

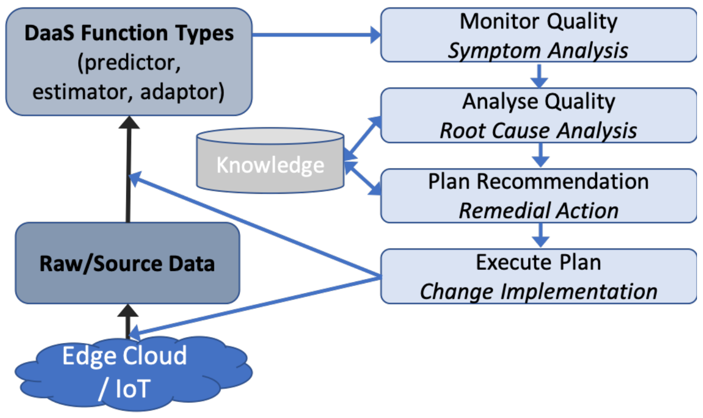


Fig. 2: Closed Service Feedback Quality Loop based on the MAPE-K pattern (see [5]).

Influencing factors for *predictor accuracy* are data incompleteness, data incorrectness, data duplication, and outdated data. An example is the count of road services per areas, which could suffer from outdated or duplicated data. For correctness, the same observations apply. For *estimator effectiveness*, an example is outdated date, which applies to self-adaptive systems for traffic control that directly depend on the current situation. *Adaptor ineffectiveness* could be caused by an incorrect format in temperature measurements (Celsius vs Fahrenheit). Some of these conditions, as in the 'outdated' case, also depend on whether the application context is stateful or stateless .

An analysis of data quality problems for an observed ML quality problem could lead to remediation recommendations [15, **?**] in two categories: *Source data:* recommendation to use other raw/source data, which could mean more, different, or less data. *ML training/testing data:* recommendation to use other ML training/testing data selected or to use even another ML technique. These would ideally be automatically derived.

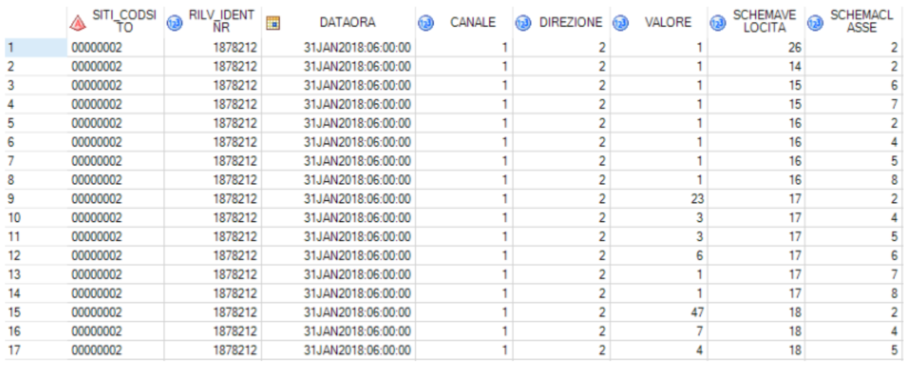
5 An IoT-Edge Traffic Management Use Case

We look now at quality assessment and symptom analysis activities.

5.1 Quality and Symptom Analysis

We identified different DaaS functions earlier that shall be looked at in terms of (1) the quality dimension and its definition, (2) a concrete example of a DaaS function and (3) the determination of quality value. In a negative quality case, we refer to the symptom. These are based on a vehicle data set based on the 'TrafficCount', combined with 'WeatherData' for the respective location, see Fig. 3. We need to consider how quality is measured and how success is determined. We distinguish the three function types and their quality goals. The functions and their expected qualities are summarised in Table 1.

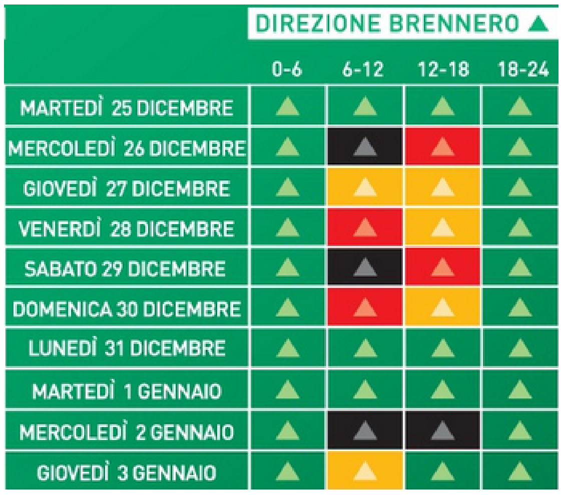
1. An *estimator* for traffic volume has the following concerns: effective and complete.



	SITI_CODSITO	RILV_IDENTNR	DATAORA	CANALE	DIREZIONE	VALORE	SCHEMAVELOCITA	SCHEMACLASSE
1	00000002	1878212	31JAN2018:06:00:00	1	2	1	26	2
2	00000002	1878212	31JAN2018:06:00:00	1	2	1	14	2
3	00000002	1878212	31JAN2018:06:00:00	1	2	1	15	6
4	00000002	1878212	31JAN2018:06:00:00	1	2	1	15	7
5	00000002	1878212	31JAN2018:06:00:00	1	2	1	16	2
6	00000002	1878212	31JAN2018:06:00:00	1	2	1	16	4
7	00000002	1878212	31JAN2018:06:00:00	1	2	1	16	5
8	00000002	1878212	31JAN2018:06:00:00	1	2	1	16	8
9	00000002	1878212	31JAN2018:06:00:00	1	2	23	17	2
10	00000002	1878212	31JAN2018:06:00:00	1	2	3	17	4
11	00000002	1878212	31JAN2018:06:00:00	1	2	3	17	5
12	00000002	1878212	31JAN2018:06:00:00	1	2	6	17	6
13	00000002	1878212	31JAN2018:06:00:00	1	2	1	17	7
14	00000002	1878212	31JAN2018:06:00:00	1	2	1	17	8
15	00000002	1878212	31JAN2018:06:00:00	1	2	47	18	2
16	00000002	1878212	31JAN2018:06:00:00	1	2	7	18	4
17	00000002	1878212	31JAN2018:06:00:00	1	2	4	18	5

Fig. 3: Traffic Count Data Set - based on Regional Recordings `https://mobility.api.opendatahub.bz.it/v2/swagger-ui.html` - (see [5])

   Effectiveness can be defined as to what extend the estimation can be correct and effective for better performance. For example, to estimate the traffic volume for an August in general irrespective of concrete weather, we obtain the result by using supervised learning. To ensure the correctness of the estimation, the historic data should be checked, e.g., the results from earlier years $Y1$ and $Y2$ imply the estimation year $Y3$, i.e., $Y1, Y2 \rightarrow Y3$. This function is used for *long-term road planning* for all roads, see Fig. 3. Completeness for the estimator is easy to calculate.
2. A *predictor* for traffic volume and level for a concrete future date is the second function.
   The function calculates a volume V using $F(T,C,W) \rightarrow V : INT$ based on temperature, number of cars and weekday. For immediate assessment, we need to check observations in the current state and assume problems might have been there also in the past. Furthermore, we cannot predict the likelihood of any source of problem. As another example, we can consider a predictor for car types: accurate and correct. In this situation correctness can be considered a special case of accuracy, that is 100% accuracy in this situation. This function is used for *short/mid-term management* on major roads as exemplified in Fig. 4.
3. An *adaptor* for traffic signs has effectiveness and optimality as concerns.
   Contrary to estimators and predictors, an adaptor also proposes some actions after the calculation and evaluation of the situation. An adaptor function should be effective. For this function the calculations for speed are done based on car volume and emissions ($F(C,E) = Speed$). The optimal target is minimal emissions E_{min}, but this is constrained by traffic throughput (too restrictive speed limit might cause traffic stop and thus low emissions, but throughput is inadequate). The evaluations are done by checking whether the goals are achieved or not. This is used for example for *immediate motorway management*, see Fig. 5. If the quality is insufficient, the problem could be either the training data and sensors.



	DIREZIONE BRENNERO ▲			
	0-6	6-12	12-18	18-24
MARTEDÌ 25 DICEMBRE	🟩	🟩	🟩	🟩
MERCOLEDÌ 26 DICEMBRE	🟩	⬛	🟥	🟩
GIOVEDÌ 27 DICEMBRE	🟩	🟨	🟨	🟩
VENERDÌ 28 DICEMBRE	🟩	🟥	🟨	🟩
SABATO 29 DICEMBRE	🟩	⬛	🟥	🟩
DOMENICA 30 DICEMBRE	🟩	🟥	🟨	🟩
LUNEDÌ 31 DICEMBRE	🟩	🟩	🟩	🟩
MARTEDÌ 1 GENNAIO	🟩	🟩	🟩	🟩
MERCOLEDÌ 2 GENNAIO	🟩	⬛	⬛	🟩
GIOVEDÌ 3 GENNAIO	🟩	🟨	🟩	🟩

Fig. 4: Public DaaS [Web Site] – Traffic Level Prediction – motivated by `https://www.autobrennero.it/en/on-the-road/traffic-forecast/` - (see [5])

Fig. 5: Public DaaS [Road Sign] – Dynamic Traffic Signpost on the Motorway - (see [5])

Table 1: Use Cases – DaaS Functions and Quality (see [5])

	(1) Estimator	(2) Predictor	(3) Adaptor
Function & Quality	Estimator: effective, complete	Predictor: accurate, correct	Adaptor: effective
Sample Function	estimate the traffic volume for an August in general	car type categorisation	calculate traffic sign action (target: change speed limits to lower emissions)
Quality Value	*Calculation*: correctness of prediction for historic data (could use for training/validation data from past August or previous July). *Success*: degree of effectiveness for threshold T	*Calculation*: Precision, Recall based on *TP, FP, FP, FN*. *Success*: a threshold T on predefined degree of accuracy.	*Calculation*: observation after applying action $OBS_E(Apply(Action))$. *Success*: is effective, if $E_{i+1} < E_i$ for emissions E. The aim is the reach a target emission while not having too slow traffic.

5.2 Root Cause Analysis and Remedial Action Recommendation

The root cause analysis shall be looked at in more detail. The use cases are summarised in the respective tables for the data quality analysis and for the problem root cause analysis and recommendation.

For all cases, we note (i) calculation of metric, (ii) assessment of problem situation, (iii) analysis of possible root causes (along the two categories or more fine-granular in terms of concrete data quality dimensions, and (iv) a strategy for better cause determination. The aim is now to determine a cause (either definitive or likely) from sources such as training data or source data.

The three main steps hall be presented in the next subsections.

5.3 Step 1 Metric Calculation and Step 2 Problem Assessment

These steps are presented in Table 2. For the predictor accuracy, we analysed the accuracy input parameters:

- *TP:* if the current state $OBS(currentstate)$ is correct and the next state
$$V = OBS(nextstate)$$
also results in correctness – indicates a given condition exists, when it really does.
- *FP:* if current state $OBS(currentstate)$ is incorrect and next state
$$V = OBS(nextstate)$$
results in correctness – indicates a given condition exists, when it does not.
- *TN:* if current state $OBS(currentstate)$ is correct and next state
$$V \neq OBS(nextstate)$$
results in incorrectness – indicates a condition does not hold, when it really does not.
- *FN:* if current state $OBS(currentstate)$ is incorrect and next state
$$V \neq OBS(nextstate)$$
results in incorrectness – indicates that a condition does not hold, while in fact it does hold.

5.4 Step 3 – Root Cause Analysis and Recommendation

Table 2 presents the use case. For Case 2 for example, false positive (FP) is an error in data reporting, in which a result improperly indicates a problem, when in reality it is not present. More concretely, this could be a vehicle that is not a car, but incorrectly recognised as one. A false negative (FN) is an error in which a result wrongly indicates no quality problem (the result is negative), when actually it is present. Here, raw sensor data can be wrong, e.g., sensors giving incomplete data (such as too small dimensions given) so that a van is recognised as a car) or training data is wrong. For the latter case, e.g., not enough annotated/labelled cars are in the training set so that very large cars (SUV) are identified as a van or truck.

Based on these observations, FP problem causes are as follows:

Table 2: Use Cases – DaaS Quality Analysis (see [5])

	(1) Estimator	**(2) Predictor**	**(3) Adaptor**
Calculation of Metric	$F(C,P) \rightarrow Volume$ estimates volumes of traffic for general periods	$F(T,C,W) \rightarrow Volume$ predicts vehicle numbers based on temperature, counted cars, weekday	$F(C,E) = Speed$ adapts speed limits based on car volume and emission
Assessment of Problem Situation	Goal achievement: – the results from earlier years $Y1$ to $Y2$ imply the estimation $Y3$, i.e., $Y1,\ldots,Y2 \rightarrow Y3$	Goal achievement: – Four cases occur: (i) 100% accuracy, (ii) $<$ 100% accuracy, but within tolerance (threshold T), (iii) $<$ T%, (iv) undefined. – Accuracy is defined using $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$.	Goal achievement: – emissions (primary): $E_{new} \leq E_T$ for threshold T as ultimate goal; $E_{new} < E_{old}$ as just improvement, i.e. these are 100% effective, and x% effective. – throughput: $OBS_C($ $Apply($ $Speed)) = C_{new}$ – secondary: C_{new} as close as possible to C_{old}

- raw data is wrong: e.g., sensors giving incomplete data such as too small dimensions given so that a van is recognised as a car,
- training data is wrong: e.g., not enough annotated/labelled cars in training set so that very large cars (SUV) are identified as vans/trucks.

Similarly, the FN problem causes can also be summarised:

- raw data wrong: either sensors giving incomplete data (e.g., too big dimensions provided, so that its recognised as a van) or sensors giving incomplete data (e.g., too small dimensions given so that a van is recognised as a car),
- training (data) wrong: not enough annotated/labelled cars in training set so that very large cars (SUV) are identified as van/truck training (data) wrong (not enough annotated/labelled cars in training set so that very small or very large cars are not covered.

Root Cause Analysis. Problems can be identified by searching for indicative patterns or anomalies. In pattern identification different situations can be distinguished.

For example, a steep decrease in a quality graph over time (time series) could point to a sudden sensor failure. A gradual decrease of quality could point to problems within the data. In a flat effectiveness quality graph, the problem could be arising from the training data. Or in other cases, in a classification function, patterns in sequences of symbols can have different meanings in each situation, e.g., unexpected repeated symbols or unexpected increase in symbols. Examples where *time series* can help are firstly *outages*, i.e., no data for a period (communications problem), and secondly *incorrect data*, i.e., sensors faulty (e.g., giving to high measurements). Here, the Assessment is based on the detection of patterns or anomalies.

A time series for a current assessment could for example be a normal series $CBTCBT$, changing into $CCTTCC$ as a sequence that shows an unusual pattern (here for vehicle categories car C, bike B and truck T). The cause analysis uses pattern/anomaly detection, with pattern mappings to the data level. Time series can be used for predictive maintenance, i.e., through the identification of changing patterns can predict a future problem.

Remedial Action Recommendation. A strategy generally used in quality remediation is training data validation. Different DaaS functions F_i are created for different training data sets and then according to the result different options can be taken.

One option is to compare functions themselves and another one is to compare input/output values. For instance, we could do majority vote on similarity (e.g., on 3 data sets). If one is different, this set has a specific property, e.g., more July data than others. The recommendation could be to check July data for completeness or accuracy. This might have to be done manually. If necessary, a different function needs to be constructed.

In this context, the primary remedial strategy is starting with training data changes and/or constructing different DaaS functions. An automated comparison can then be carried out, in relation to historic data or between different functions.

5.5 Transferability

We looked at IoT settings, based on sensors as data producers. In that context, we have used traffic and weather data sets.

We considered these two domains – traffic management and weather – so far as examples of discrete numeric data. Another application domain is mobile learning that equally includes heavy use of data being collected from and delivered to mobile learners and their devices and includes the usage of multimedia content being delivered to mobile learners and their devices [21, 35, 30, 28, 22]. These systems also rely on close interaction with semantic processing of interactions in order to support cognitive learning processes [16, 19], which would help to increase the understandability of the DaaS offering provided. Here the setting is different in that multimedia content is produced and transferred. This can equally cause incompleteness and incorrectness problems, but here the differences is that continuous streams of binary data is affected.

A further direction is the implementation of self-adaptive ML quality management in an IoT-edge continuum [51, 29]. In that context, similar to the original setting, sensors produce data (albeit sensors measuring often virtualized infrastructure performance) that is after some analysed translated into instructions to be executed by actuators in the same IoT edge space. Here, video surveillance could be considered, where cameras monitor for example building that need protection. Image quality could be monitored and faults in the video or transmission system detected. Microservices and containers [46, 47, 34, 38] would here be the main artifacts that would be monitored, causing continuously produced input data.

From our results, we can ascertain that sensor faults have a different impact than for instance network failures and that with some certainty a defect cause can be identified. This, however, needs to be further explored and confirmed for other data than the numerical and limited volume situations considered here.

6 Related Work

In our review of related work, we look at the data quality level, machine learning process perspective and DaaS model quality. Data-level quality has been considered in [33], [10], [43]. In [33], data quality problems have been classified into two groups of context-independent and context-dependant distinguished by the data and user perspective. In [10], an architecture based on Blockchain technology has been proposed to improve data quality and false data detection. In [43], a cross-domain prototype of a lightweight distributed architecture for IoT was also presented, providing supporting algorithms for the assessment of data quality and security. In order to adapt to our IoT application context, we build on [33].

A machine learning workflow with nine stages was presented in [2], where the early stages are data oriented. Generally, workflows connected to ML are non-linear and often contain several feedback loops to earlier stages. If a system contains multiple machine learning components, that interact together in complex and unexpected ways, this type of workflow can become complex. Our aim is to investigate here as a novel perspective a broader loop from the later final ML function stages to the initial data and ML training configuration stages. There is also work specific to the machine learning layer, such as [39] and [9]. There, different supervised learning approaches were used. They noted that different methods have different applications. They analysed in this context the effect of calibrating these models via Platt scaling and isotonic regression on their performance as a quality concern.

Specific quality metrics applied to ML-based data models have been investigated. FOr instance, [23] discusses the area under the receiver operating characteristic curve (AUC) as a sample quality for classification models. In [45], a solution for model governance in production machine learning is introduced where provenance information and be traced demonstration the emergence of an ML prediction solution. Also the quality of data in ML has been looked at. An application use case was presented there, but without a systematic coverage of quality aspects.

Data quality is important in many ML-supported DaaS applications, such as scientific computing. In [12], a high-energy physics experiment as an IoT-type setting is investigated, which demonstrates the need for a systematic, automated approach to achieve a higher level of accuracy compared to training problems arising from manual data labelling. In [11] another IoT and edge cloud setting is considered that emphasises the uncertainty of sensory data as the main problem causes. Their proposal is to give data quality a crucial role in this process. [14] is related to our setting, but only covers IoT root causes in the analysis, but does not cover ML training data problems. Our aim is to condense different individual quality concerns in a single quality model for DaaS data services that takes on board lessons learned from [12, 11, 14], but also closed feedback loop as we indicated in our architecture model.

In [44], the authors employed three supervised machine learning (ML) algorithms, Bayesian Networks, Decision Trees and Multilayer Perceptrons for the flow-based classification of six different types of Internet traffic including peer-to-peer(P2P) and content delivery (Akamai) traffic. They investigated the dependency of the traffic classification performance on the amount and composition of training data. They also showed that ML algorithms such as Bayesian Networks and Decision Trees are suitable for Internet traffic flow classification at a high speed and proved to be robust with respect to applications that dynamically change their source ports.

[24] and [6] employed machine learning methods to improve accuracy. Improving the data accuracy and reliability is one of our main goals which makes these papers a good reference for us that provide accuracy specific quality improvement strategies. In [8] a survey of data mining applications in business is provided to investigate the use of learning techniques. [31] provides an overview of the application of ML to optical communications and networking. [25] proposes a deep learning method that combines CNN and LSTM to detect abnormal network traffic, especially unknown intrusions. In [2], nine stages were specified for a ML workflow in which some of the stages were data oriented. ML workflows are highly non-linear and contain several feedback loops which may loop back to any previous stage. This workflow can become even more complex if the system is integrative, containing multiple ML components which interact together. [8], [31], [25] and [2] have all presented useful background and information on how to use machine learning for different purposes and in different applications. All this information is needed for us in order to use machine learning and investigate specific quality concerns in our case study application.

The machine learning model layer has been studied in [39] and [9]. Different algorithms and approaches were introduced and used in these papers which were mostly building on supervised learning algorithms. They also examined the effect that calibrating the models via Platt scaling and isotonic regression has on their performances and quality. [1] introduced an automated secure continuous cloud service availability framework for smart connected vehicles that enables an intrusion detection mechanism against security attacks and provides services that meet users' quality of service (QoS) and quality of experience (QoE) requirements.

Intrusion detection is accomplished through a three-phase data traffic analysis, reduction, and classification technique used to identify positive trusted service requests against false requests that may occur during intrusion attacks. The solution adopts deep belief and decision tree machine learning mechanisms used for data reduction and classification purposes, respectively. The framework is validated through simulations to demonstrate the effectiveness of the solution in terms of intrusion attack detection. We plan to expand what the authors in this paper did for our project. This is a relevant example of how to improve machine learning model quality. The papers in this section mostly investigate new methods for a better accuracy in data using different machine learning methods.

The quality of data in a machine learning approach has been investigated in [14], where an application use case is presented that identifies relevant data quality concerns in an IoT setting. A systematic coverage of quality aspects, is however not attempted. [40] and [18] discusses the ensuring of fairness in machine learning to advance health equity which is a concern that is different from the usual accuracy.

Table 3: DaaS Quality Assessment Dimensions (see [5])

	DaaS Quality Value		
quality value	accuracy	correct/ effective:	optimal
metric & measurement	mostly done manual, maybe automated with other sensors, e.g., optical issues (dust) or loss of connectivity can be detected	historic data – can be mostly automated	can be automated, but needs waiting for the next state; can either be ML data or raw data

Table 4: DaaS Quality Assessment Dimensions (see [5])

	DaaS Quality Time Series		
quality value	accuracy	correct/ effective	optimal
metric & measurement	determine source by mapping time series to underlying raw data sequences (e.g., car type series)	temperature prediction series (jump $>$ 20 degrees is sensor fault)	time series could be difficult to interpret (if heating switched on or cloud workload is suddenly high), the adaptor will struggle

7 Conclusions

The aim of data processing solution is to make data accessible that in its original source data format would not be useful to a consumer. Machine learning is often used to process this data in order to create meaningful information for the final DaaS consumer. Normally, accuracy is the key objective of the created data models, but we aim here at a broader categorisation of qualities, covering the source data as well as the ML-based DaaS model layer. We presented here an integrated DaaS quality management framework. We provided a fine-granular model for a range of service quality concerns addressing common types of machine learning function types such as predictors, estimators and adaptors.

The central technical contribution of our paper is the mapping of observable quality deficiencies of DaaS functions to underlying, possibly hidden data quality problems.

Providing a root cause analysis for symptoms observed by the service consumer is the objective. In addition, remedial actions for the identified problems and their causes can be proposed to the user by the solution.

We validated of both DaaS function types and related data quality types in symptom and root cause analysis through use cases. In our IoT and edge cloud case study, quality data regarding current situations and also time series have been addressed. Furthermore, *aggregation* is a mechanism based on location or time, but, this has not been covered in our use cases. Both situational and time series-based quality analysis have been covered in Tables 3 and 4, respectively.

As part of our future work, some open concerns shall be addressed. We have introduced here informal definitions for function and data quality, but these still need to be formalised in our framework. Furthermore, we will also deal with more complex settings with multiple clusters of data producers that would need to be coordinated, following our earlier work on these architectures [17, 42, 51, 50]. Thus work would allow us to generalise the results to multiple data services [37] in an edge cloud setting. A final concern is root cause identification, e.g., basing this on the analysis of ML techniques like regression, classification or clustering. Also statistical/probabilistic models such as Markov Models can be used to map observable data processing function quality to hidden data quality.

Acknowledgements

This work has been performed partly within a Ph.D. Programme funded through a bursary by the Südtiroler Informatik AG (SIAG).

References

1. Moayad Aloqaily, Safa Otoum, Ismaeel Al Ridhawi, and Yaser Jararweh. An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Networks*, 90, 02 2019.
2. Saleema Amershi, Andrew Begel, Christian Bird, Rob DeLine, Harald Gall, Ece Kamar, Nachi Nagappan, Besmira Nushi, and Tom Zimmermann. Software engineering for machine learning: A case study. In *Intl Conf on Software Engineering - Software Engineering in Practice track*. IEEE, May 2019.
3. Shelernaz Azimi and Claus Pahl. A layered quality framework in machine learning driven data and information models. In *22nd International Conference on Enterprise Information Systems*, 2020.
4. Shelernaz Azimi and Claus Pahl. Root cause analysis and remediation for quality and value improvement in machine learning driven information models. In *22nd International Conference on Enterprise Information Systems*, 2020.
5. Shelernaz Azimi and Claus Pahl. Continuous data quality management for machine learning based data-as-a-service architectures. In *International Conference on Cloud Computing and Services Science CLOSER*, 2021.
6. Paola Bermolen, Marco Mellia, Michela Meo, Dario Rossi, and Silvio Valenti. Abacus: Accurate behavioral classification of p2p-tv traffic. *Computer Networks*, 55(6):1394 – 1411, 2011.

7. Jan Bosch, Helena Holmström Olsson, and Ivica Crnkovic. Engineering ai systems: A research agenda. In *Artificial Intelligence Paradigms for Smart Cyber-Physical Systems*, pages 1–19. IGI Global, 2021.
8. Indranil Bose and Radha Mahapatra. Business data mining - a machine learning perspective. *Information Management*, 39:211–225, 12 2001.
9. Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, page 161–168, 2006.
10. Roberto Casado-Vara, Fernando de la Prieta, Javier Prieto, and Juan M. Corchado. Blockchain framework for iot data quality via edge computing. In *Proceedings of the 1st Workshop on Blockchain-Enabled Networked Sensor Systems*, page 19–24, 2018.
11. J. De Hoog, S. Mercelis, and P. Hellinckx. Improving machine learning-based decision-making through inclusion of data quality. *CEUR Workshop Proceedings*, 2491, 2019.
12. K. Deja. Using machine learning techniques for data quality monitoring in cms and alice. *Proceedings of Science*, 350, 2019.
13. Bradley Efron. Prediction, estimation, and attribution. *Journal of the American Statistical Association*, 115(530):636–655, 2020.
14. Lisa Ehrlinger, Verena Haunschmid, Davide Palazzini, and Christian Lettner. A daql to monitor data quality in machine learning applications. In *Database and Expert Systems Applications*, pages 227–237, 2019.
15. Daren Fang, Xiaodong Liu, Imed Romdhani, Pooyan Jamshidi, and Claus Pahl. An agility-oriented and fuzziness-embedded semantic model for collaborative cloud service search, retrieval and recommendation. *Future Generation Computer Systems*, 56:11–26, 2016.
16. Daren Fang, Xiaodong Liu, Imed Romdhani, Pooyan Jamshidi, and Claus Pahl. An agility-oriented and fuzziness-embedded semantic model for collaborative cloud service search, retrieval and recommendation. *Future Gener. Comput. Syst.*, 56:11–26, 2016.
17. Frank Fowley, Claus Pahl, Pooyan Jamshidi, Daren Fang, and Xiaodong Liu. A classification and comparison framework for cloud service brokerage architectures. *IEEE Trans. Cloud Comput.*, 6(2):358–371, 2018.
18. L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang. Cost efficient resource management in fog computing supported medical cyber-physical system. *IEEE Transactions on Emerging Topics in Computing*, 5(1):108–119, Jan 2017.
19. Muhammad Javed, Yalemisew M. Abgaz, and Claus Pahl. Ontology change management and identification of change patterns. *J. Data Semant.*, 2(2-3):119–143, 2013.
20. J. Jiarpakdee, C. Tantithamthavorn, H. K. Dam, and J. Grundy. An empirical study of model-agnostic techniques for defect prediction models. *IEEE Transactions on Software Engineering*, pages 1–1, 2020.
21. Claire Kenny and Claus Pahl. Automated tutoring for a database skills training environment. In *36th Technical Symposium on Computer Science Education, SIGCSE*, pages 58–62. ACM, 2005.
22. Claire Kenny and Claus Pahl. Automated tutoring for a database skills training environment. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '05, page 58–62, New York, NY, USA, 2005. Association for Computing Machinery.
23. Ross Kleiman and David Page. Aucμ: A performance metric for multi-class machine learning models. In *Intl Conference on Machine Learning*, pages 3439–3447, 2019.
24. Wei Li and Andrew Moore. A machine learning approach for efficient traffic classification. pages 310 – 317, 11 2007.
25. Xianglin Lu, Pengju Liu, and Jiayi Lin. Network traffic anomaly detection based on information gain and deep learning. pages 11–15, 04 2019.

26. Mohammad Saeid Mahdavinejad, Mohammadreza Rezvan, Mohammadamin Barekatain, Peyman Adibi, Payam Barnaghi, and Amit P. Sheth. Machine learning for internet of things data analysis: a survey. *Digital Communications and Networks*, 4(3):161 – 175, 2018.
27. Milen S. Marev, Ernesto Compatangelo, and Wamberto Weber Vasconcelos. Towards a context-dependent numerical data quality evaluation framework. *CoRR*, abs/1810.09399, 2018.
28. Mark Melia and Claus Pahl. Constraint-based validation of adaptive e-learning courseware. *IEEE Trans. Learn. Technol.*, 2(1):37–49, 2009.
29. Nabor Chagas Mendonça, Pooyan Jamshidi, David Garlan, and Claus Pahl. Developing self-adaptive microservice systems: Challenges and directions. *IEEE Softw.*, 38(2):70–79, 2021.
30. Sarah Murray, James Ryan, and Claus Pahl. Tool-mediated cognitive apprenticeship approach for a computer engineering course. In *International Conference on Advanced Learning Technologies*, pages 2–6. IEEE, 2003.
31. F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore. An overview on application of machine learning techniques in optical networks. *IEEE Communications Surveys Tutorials*, 21(2):1383–1408, 2019.
32. Thuan L Nguyen. A framework for five big v's of big data and organizational culture in firms. In *International Conference on Big Data*.
33. Tony O'Brien, Markus Helfert, and Arun Sukumar. The value of good data- a quality perspective a framework and discussion. In *International Conference on Enterprise Information Systems*, 2013.
34. Claus Pahl. An ontology for software component matching. In *International Conference on Fundamental Approaches to Software Engineering*, pages 6–21. Springer, 2003.
35. Claus Pahl, Ronan Barrett, and Claire Kenny. Supporting active database learning and training through interactive multimedia. In *9th Conference on Innovation and Technology in Computer Science Education, ITiCSE*, pages 27–31. ACM, 2004.
36. Claus Pahl, Ilenia Fronza, Nabil El Ioini, and Hamid R. Barzegar. A review of architectural principles and patterns for distributed mobile information systems. In *International Conference on Web Information Systems and Technologies*.
37. Claus Pahl, Nabil El Ioini, Sven Helmer, and Brian A. Lee. An architecture pattern for trusted orchestration in iot edge clouds. In *Intl Conf on Fog and Mobile Edge Computing*. IEEE, 2018.
38. Claus Pahl, Pooyan Jamshidi, and Olaf Zimmermann. Microservices and containers. *Software Engineering 2020*, 2020.
39. Dariusz Plewczynski, Stéphane A. H. Spieser, and Uwe Koch. Assessing different classification methods for virtual screening. *Journal of Chemical Information and Modeling*, 46(3):1098–1106, 2006.
40. Alvin Rajkomar, Michaela Hardt, Michael D Howell, Greg Corrado, and Marshall H Chin. Ensuring fairness in machine learning to advance health equity. *Annals of internal medicine*, 169(12):866–872, 2018.
41. Barna Saha and Divesh Srivastava. Data quality: The other face of big data. In *2014 IEEE 30th International Conference on Data Engineering*, pages 1294–1297. IEEE, 2014.
42. Remo Scolati, Ilenia Fronza, Nabil El Ioini, Areeg Samir, and Claus Pahl. A containerized big data streaming architecture for edge cloud computing on clustered single-board devices. In *Intl Conf on Cloud Computing and Services Science*, 2019.
43. Sabrina Sicari, Alessandra Rizzardi, Daniele Miorandi, Cinzia Cappiello, and Alberto Coen-Porisini. A secure and quality-aware prototypical architecture for the internet of things. *Information Systems*, 58:43 – 55, 2016.
44. Murat Soysal and Ece Guran Schmidt. Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison. *Performance Evaluation*, 67(6):451 – 467, 2010.

45. Vinay Sridhar, Sriram Subramanian, Dulcardo Arteaga, Swaminathan Sundararaman, Drew S. Roselli, and Nisha Talagala. Model governance: Reducing the anarchy of production ml. In *USENIX Annual Technical Conference*, 2018.
46. Davide Taibi, Valentina Lenarduzzi, and Claus Pahl. Continuous architecting with microservices and devops: A systematic mapping study. In *International Conference on Cloud Computing and Services Science*, pages 126–151. Springer, 2018.
47. Davide Taibi, Valentina Lenarduzzi, Claus Pahl, and Andrea Janes. Microservices in agile software development: a workshop-based study into issues, advantages, and disadvantages. In *Proceedings of the XP2017 Scientific Workshops*, pages 1–5, 2017.
48. Chakkrit Tantithamthavorn, Jirayus Jiarpakdee, and John Grundy. Explainable ai for software engineering. *arXiv preprint arXiv:2012.01614*, 2020.
49. Somasekhar Thatipamula. Data done right: 6 dimensions of data quality. `https://smartbridge.com/data-done-right-6- dimensions-of-data-quality/`, Aug 2013. Accessed on 2021-01-16.
50. David von Leon, Lorenzo Miori, Julian Sanin, Nabil El Ioini, Sven Helmer, and Claus Pahl. A performance exploration of architectural options for a middleware for decentralised lightweight edge cloud architectures. In *Intl Conf on Internet of Things, Big Data and Security*, pages 73–84, 2018.
51. David von Leon, Lorenzo Miori, Julian Sanin, Nabil El Ioini, Sven Helmer, and Claus Pahl. A lightweight container middleware for edge cloud architectures. In *Fog and Edge Computing*, Wiley Series on Parallel and Distributed Computing, pages 145–170. Wiley, 2019.