IoT Data Validation

Best Practice Guide

***Mads Johansen**, Data Specialist,*
IoT, Data & Services Innovation, FORCE Technology (majh@forcetechnology.com)

***Anders Mynster**, (Until August 2022) Head of Department,*
IoT, Data & Services Innovation, FORCE Technology

***Alexandre Alapetite**, Principal Software Solutions Architect,*
AI & Data Analytics Lab, Alexandra Institute (alexandre.alapetite@alexandra.dk)

***Anders Depner**, (Until May 2022) Specialist,*
IoT, Data & Services Innovation, FORCE Technology

Table of Contents

1 Executive Summary

The present whitepaper describes the best practices for data validation. Data validation refers to the process of ensuring the accuracy and quality of data. It is implemented by building several checks into a system or report to ensure the logical consistency of input and stored data.

The importance of data validation cannot be understated. In the McKinsey report, "The Internet-of-Things catching up to an accelerating opportunity", Nov 2021, it is stated: *"We estimate the total value captured by the end of 2020 ($1.6 trillion), while considerable, to be in the lower end of the range of the scenarios we mapped out in 2015."* Hence, if we do not ensure that data is valid to make data driven decisions, we risk $1.6 trillion on wrong decisions. When the Danish Agency for Digital Government midway evaluated the AI signature projects for the key challenge of implementing AI for the public domain, the answer across a $30 million portfolio was: Lack of data quality.

In this paper, the best practices of the industry are presented for how to validate IoT data. The organizational challenges are presented in the section "The nature of data validation". Next, how a process can be implemented to improve the level of data quality through data validation, then tools for structuring and prioritizing the process of data validation are discussed, before presenting practical tools which can be, and are being implemented, in operational systems.

It is evident that data validation can easily become a time-consuming task and therefore it is important to reiterate that the most essential task of data validation is the definition of when data is fit-for-purpose. Is the quality and accuracy of the data sufficient to be used for data-driven decisions? Determining this as well as adopting a very critical prioritization of activities in the iterative process, can substantially improve the validity and quality of the data with minimum effort.

2 Introduction

In this whitepaper, we will present the best practices, a process, and tools for IoT data validation. This knowledge is collected and/or developed by FORCE Technology in collaboration with the Alexandra Institute, under the Nordic IoT Centre.

The data validation process is primarily focused on the validation of data from IoT systems, but the data validation process will naturally be applicable to other types of data as well, as IoT systems rarely deal with only "IoT data", which by most is seen as time-series measurements. Data validation is an iterative process that aims at ensuring a suitable level of data quality for a system or an application.

The importance of data validation cannot be understated. In the McKinsey report "The Internet-of-Things catching up to an accelerating opportunity", Nov 2021, it is stated: "We *estimate the total value captured by the end of 2020 ($1.6 trillion), while considerable, to be in the lower end of the range of the scenarios we mapped out in 2015."* Hence, if we do not ensure that data is valid to make the data driven decisions, we risk $1.6 trillion on wrong decisions. When the Danish Agency for Digital Government midway evaluated the AI signature projects for the key challenge of implementing AI for the public domain, the answer across a $30 million portfolio was: Lack of Data quality.

In the landscape of data management and data governance many aspects are included, one of these is data quality. Often companies tend to invest in data management software, only to find that the data quality is not at a level required for the data to be fit-for-purpose. In other words, the quality of the used data is not high enough to fulfil the need or the task for which it is used.

Data validation is the act of performing good practices to ensure that the data quality complies with the intended need. FORCE Technology & Alexandra Institute have developed a process and a toolset that enables an organization to perform data validation to ensure, that the required data quality is achieved.

We would like to thank Maersk, Novo Nordisk, NNE, Aarhus University, Vitani Energy Systems, and all the others who has contributed to gaining insights into their best practice recommendations.

2.1 Data Validation Process

Our data validation process is inspired by the *Data Validation Process Life Cycle* from the paper "Methodology for data validation 1.0" [Marco Di Zio et al., 2016]. It goes through an iterative process, that starts by designing a validation suite (an action plan, that comprise a series of tasks and tools, to apply to the data in order to improve the quality of the dataset). The validation suite is implemented by applying it to sample data, and refining the tasks and tools, before executing the validation suite on the real data. The results from applying the validation suite on the real data is evaluated, and the evaluation, which is compared to any feedback from stakeholders, will be used for the next iteration which is started by designing a new validation suite.

On this basis, FORCE Technology has developed a seven-step process. This seven-step process is a series of specific steps and actions to follow, to perform a successful data validation. Step one to three serves as a thorough review of the system, whereas steps four to seven is the iterative part of the data validation.

The seven steps, as seen in Figure 1, have been divided into two main sections: The "IoT System Analysis" and the "Validation Process". The following chapter will go into more detail with each step.

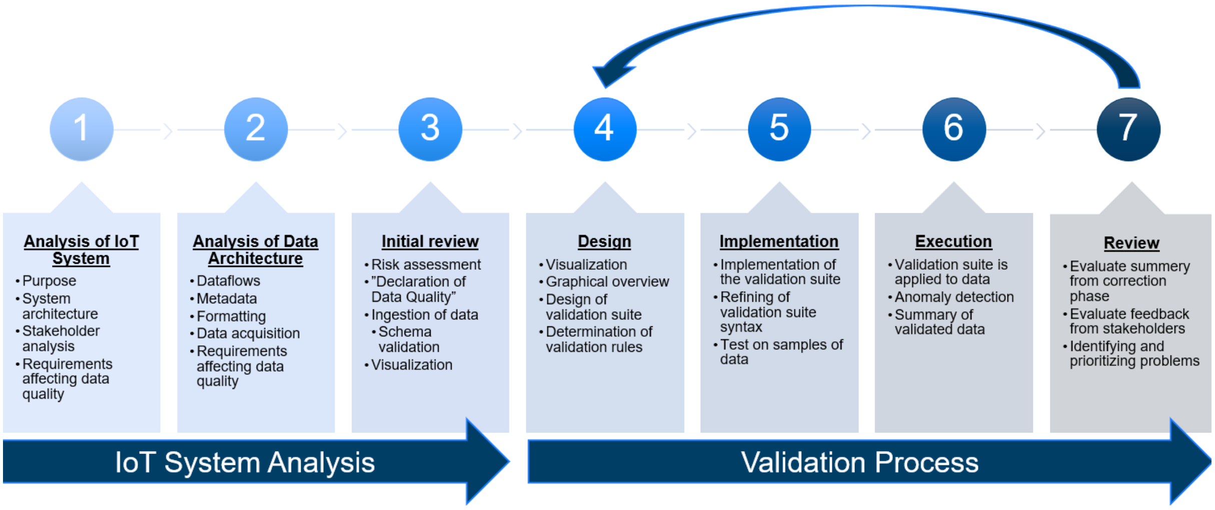

1
2
3
4
5
6
7
Analysis of IoT System
• Purpose
• System architecture
• Stakeholder analysis
• Requirements affecting data quality
Analysis of Data Architecture
• Dataflows
• Metadata
• Formatting
• Data acquisition
• Requirements affecting data quality
Initial review
• Risk assessment
• "Declaration of Data Quality"
• Ingestion of data
• Schema validation
• Visualization
Design
• Visualization
• Graphical overview
• Design of validation suite
• Determination of validation rules
Implementation
• Implementation of the validation suite
• Refining of validation suite syntax
• Test on samples of data
Execution
• Validation suite is applied to data
• Anomaly detection
• Summary of validated data
Review
• Evaluate summery from correction phase
• Evaluate feedback from stakeholders
• Identifying and prioritizing problems
IoT System Analysis
Validation Process

Figure 1: The FORCE Technology IoT Data Validation Process

3 The Nature of Data validation

"Some data is better than no data", is a statement often heard when implementing IoT solutions and starting to collect data about a given problem. However, the sentence should be completed with *"… if the data provides reliable insight about the challenge the system is trying to solve".*

Table 1 shows an example of a dataset, provided free of charge:

Time stamp	**T_1**	**T_2**
0100002022	50	45
0104042022	24	34
0104042022	24	34
0105052022	26	32
0200242022	30	30

Table 1: Example of a dataset.

Now you have "some data" but is it better than before? Probably not. Without proper descriptions of the origin of the data and how much the data can be trusted, the value is very little. Why are the time stamps irregular, and what time zone does it refer to, where is the data in T_1 and T_2 coming from and what do they indicate? The example may seem far-fetched but is unfortunately very common. To ensure that data is valuable we must validate the data, to **ensure the data is fit-for-purpose**, i.e., that it provides reliable insights about the challenge the system is trying to solve.

As described in "Data Management Body of Knowledge" [Deborah Henderson et al., 2017]. It is very common to implement IoT solutions as the following:

1. The organization purchases or develops a system that can collect and store data.
2. Once the organization starts to use the system, data is gathered, but more and more challenges with data quality occur.

3. A disciplined practice for managing data quality and performing data validation is introduced.
4. The organization leverages the benefits of well managed data.

As experts in data validation, it is tempting to think that every solution should start with a data governance policy, procedures to validate data and tools for automating this. However, the reality is that this would typically lead to excessive use of resources to do so, draining the resources from the development of the actual system. Hence, there is a need to find a good balance in doing the right amount of data validation and management for the challenge(s) being solved. Therefore, the four-step best practice, outlined above, is typically a very good approach to building the system and achieving proper data quality.

In this whitepaper, we have gathered insights from some of the leading organizations within IoT and data management in Denmark, to describe how they approach data validation, and to describe their best practices when doing so.

3.1 Who is performing the data validation?

Very often, the people validating the data are not the same as those producing the data. Sometimes they are in the same organization, but as data start to flow more and more between organizations, the people validating the data will often also be placed in an external organization.

It is essential that the validation is not seen as an obstacle, but as help to the organization or unit producing the data. In case of too rigid or strict data validation, the data will not be produced or shared. Therefore, the people performing the data validation and the processes they have implemented, need to be approachable and flexible. A friendly guide to obtaining better data quality.

3.2 Upstream and downstream workload

Data validation shifts workload between data upstream and data downstream, i.e., between those generating data and those using data. As the number of data consumers grows, the combined workload also increases with a lack of data quality. As the number of data producers grows, the workload increases with more data validation. Hence, increasing data validation decreases workload for the data users but increases for the data generators.

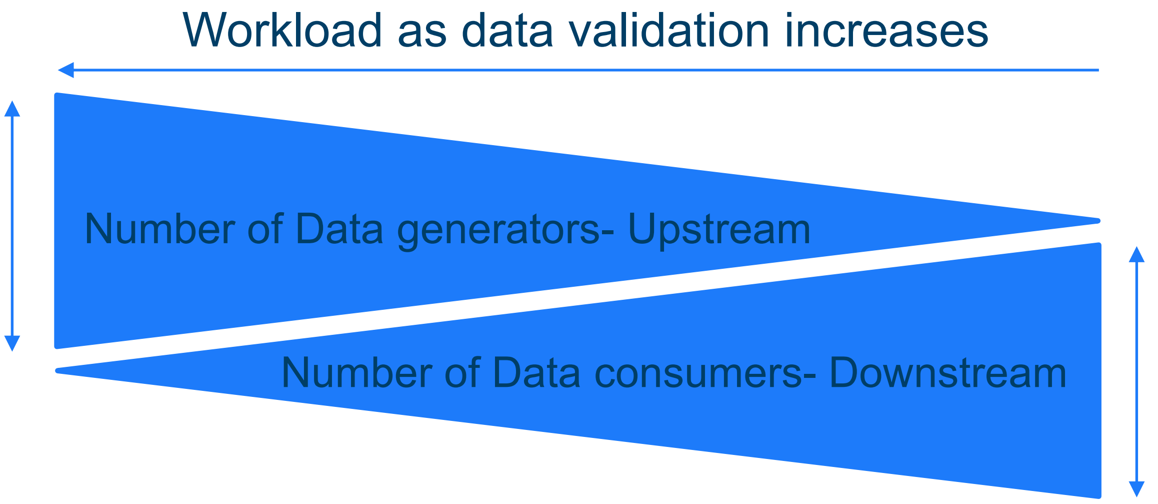

Workload as data validation increases
Number of Data generators- Upstream
Number of Data consumers- Downstream

Figure 2: Workload distribution depending on number of data generators and data consumers.

Therefore, before deciding on how much work should be put into data validation, it is advisable to know how many data generators and data consumers there are. If there are many generators, the increased requirements of data validation will significantly increase the workload, but if there are only a few users, the combined workload will not decrease much, as depicted in the figure below.

Remember to consider, how this distribution will change within a foreseeable timeframe.

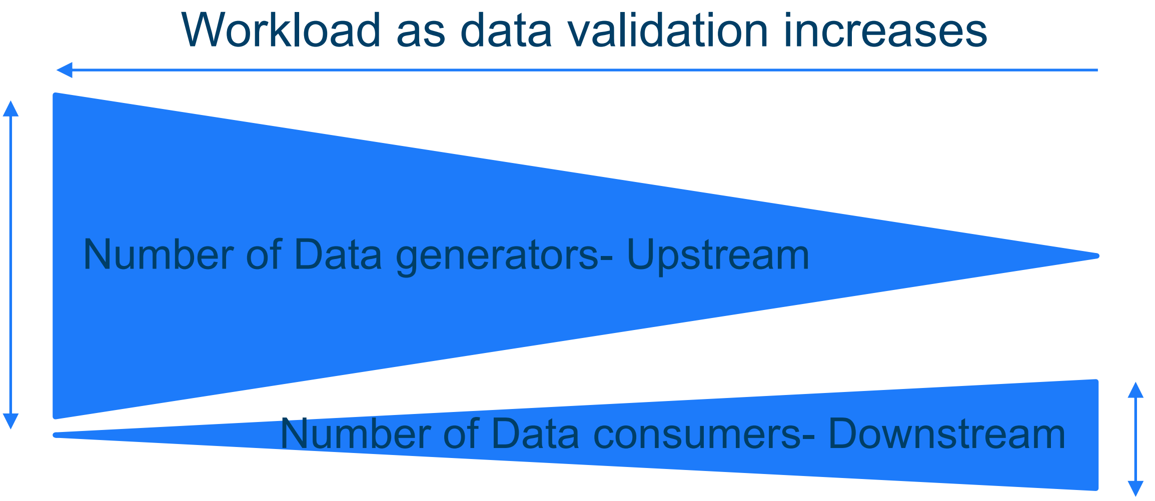

Workload as data validation increases
Number of Data generators- Upstream
Number of Data consumers- Downstream

Figure 3: Workload distribution depending on number of data generators and data consumers as the number of generators grow and consumers are reduced.

3.3 Most valuable sensors and Data

Many systems contain a large amount of data from many data sources and to ensure maximum data quality in all of these data sources requires an excessive workload. Therefore, the MVP data sources and sensors should be defined. These are sensors that are essential for fulfilling the purpose of the system. Most organizations define these in a meeting with the most important internal and external stakeholders, where the list of MVP sensors is compiled. The list is then reviewed yearly. The sensors and data sources in the list are then being monitored more rigorously than the rest of the sensors.

This can be done for instance by using the SHAP tool for Python, which can rank how influential individual data sources are on the output. It is primarily intended for machine learning purposes, but matched with proper subject matter expertise, it can provide a good starting point for the list with critical data sources.

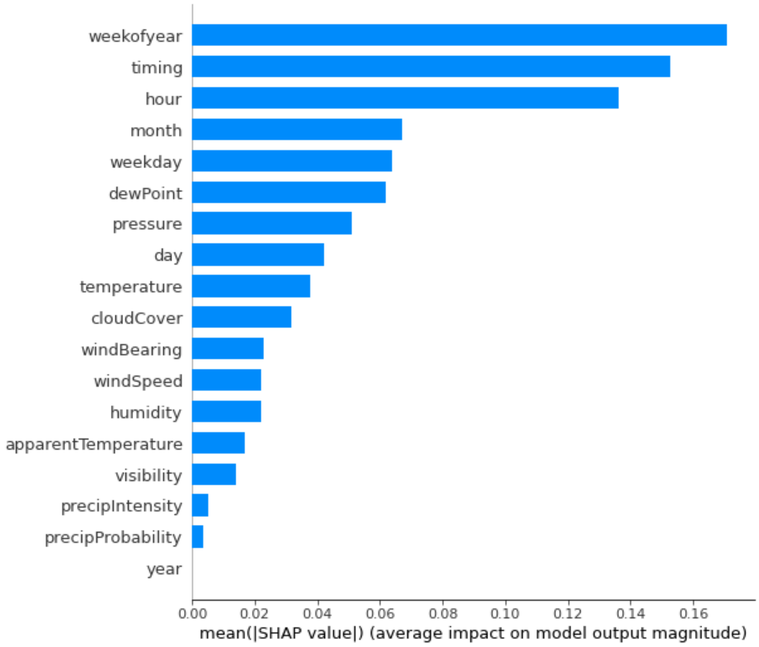


Figure 4: Example of ranking the most impactful data sources when predicting the energy consumption of a house.

3.4 Subject matter expertise

Subject matter experts are people who have expertise within the topic that the system is measuring on – sometimes also referred to as domain experts. This could be machine operators, repair technicians, doctors, nurses, etc. depending on the system focus. As a rule of thumb, the big companies use one subject matter expert per three developers.

If many subject matter experts are available, then the amount of data needed for solving the challenge, is less. If few or no subject matter experts are available, then the amount of data grows inverse exponentially. This is essential to understand, since having data available to non-subject matter experts also increases the requirements to the data quality – as they have less experience in identifying errors in the data, and for this reason the descriptions (metadata) need to be more detailed to avoid misunderstandings.

It has to be pointed out that, if you have no subject matter experts or very little access to them, the data validation goals can become purely academic, and the conclusions on data, even though it is validated, can be very wrong. An example of this was a company who had data scientists to perform an analysis of operational efficiency of machines vs the amount of time spent on maintenance. Having crunched the data, the conclusion was that the company should not perform maintenance at all! The subject matter experts looked baffled at the data and quickly found, that the reason for the conclusion was that the data consisted of new and old machines. The new had not yet had maintenance performed and were also the best performing – hence a logical conclusion, but knowing that machines break if not maintained, the company did not follow the advice.

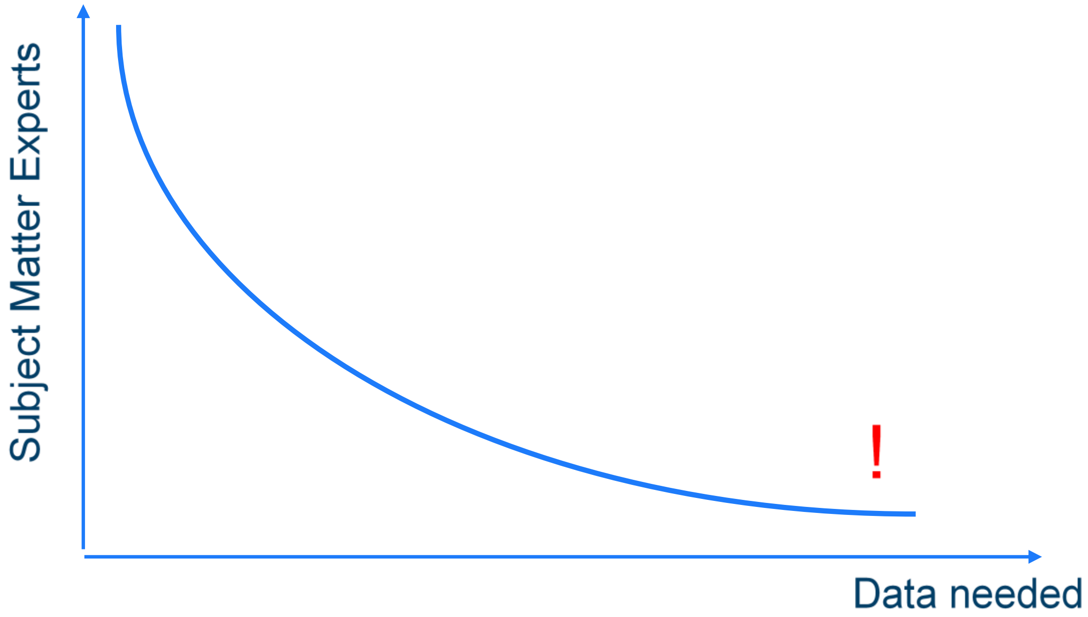


Figure 5: Subject matter expertise vs the needed data to gain insights to the system. Be aware if you do not have subject matter experts, wrong conclusions can be drawn by only looking at data.

3.5 Manual, Semi-Automated and Automated

It is very tempting to think, that there is a need for automating every process for data validation. However, automation is no trivial task, and it is time consuming to get it right. Therefore, the best practice guide is that some problems are better solved initially as manual processes – the important thing is that they are performed regularly. Key questions to address are:

- Who does it?
- What should be done?
- How often?

After the issue has been solved 10 times or more, the data is sufficient for automation. In addition, it becomes possible to evaluate the workload per month for fixing this item manually and how much it will require to automate. Here the dimension of time criticality can also be evaluated: Is this a data error that needs to be fixed within days or seconds?

A few examples could be:

- Validation of rainfall sensor data by comparison to meteorological reference data.
- Automated safe state of the system when data is detected as corrupt.
- Visual inspection of data on a weekly basis to discover unknown data errors.

3.6 Design of experiments – exploring the outcomes

When looking at the data in Figure 6, there is no apparent structure to the data and hence the validity appears poor. It is very hard to see if there is a pattern.

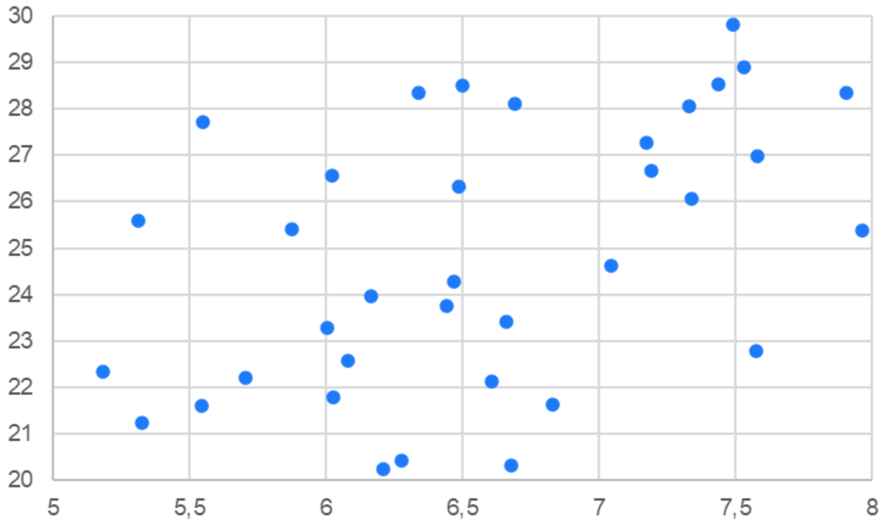


Figure 6: Data obtained from a system in operation.

However, when increasing the range of the data, a clear trend can be seen in Figure 7. Often when IoT systems are measuring on processes, the processes do not go to the extremes and therefore only a small section of the outcomes can be seen. It is essential to design experiments to fill the outcome space, to determine the relations between inputs and outputs. Otherwise, the relations between inputs and outputs can be hard to determine. The complexity increases as the number of inputs and outputs of the system increase.

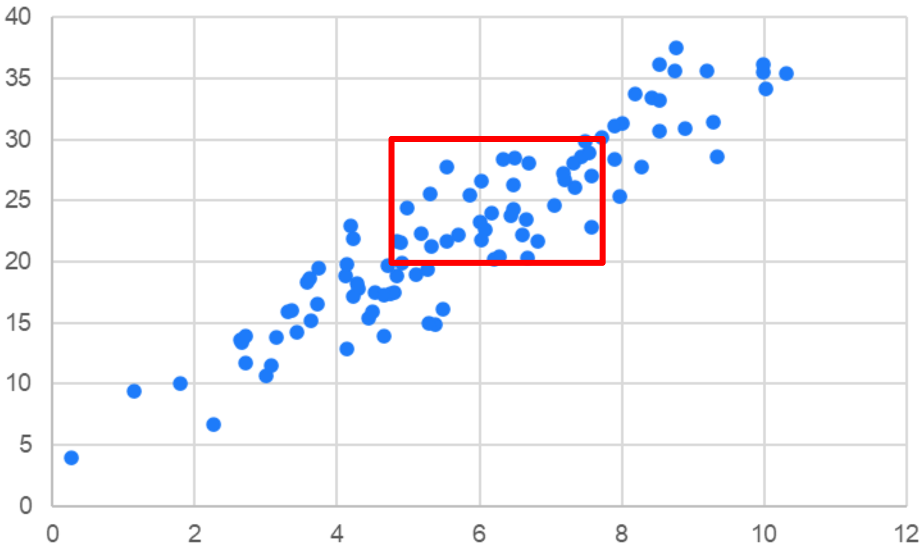


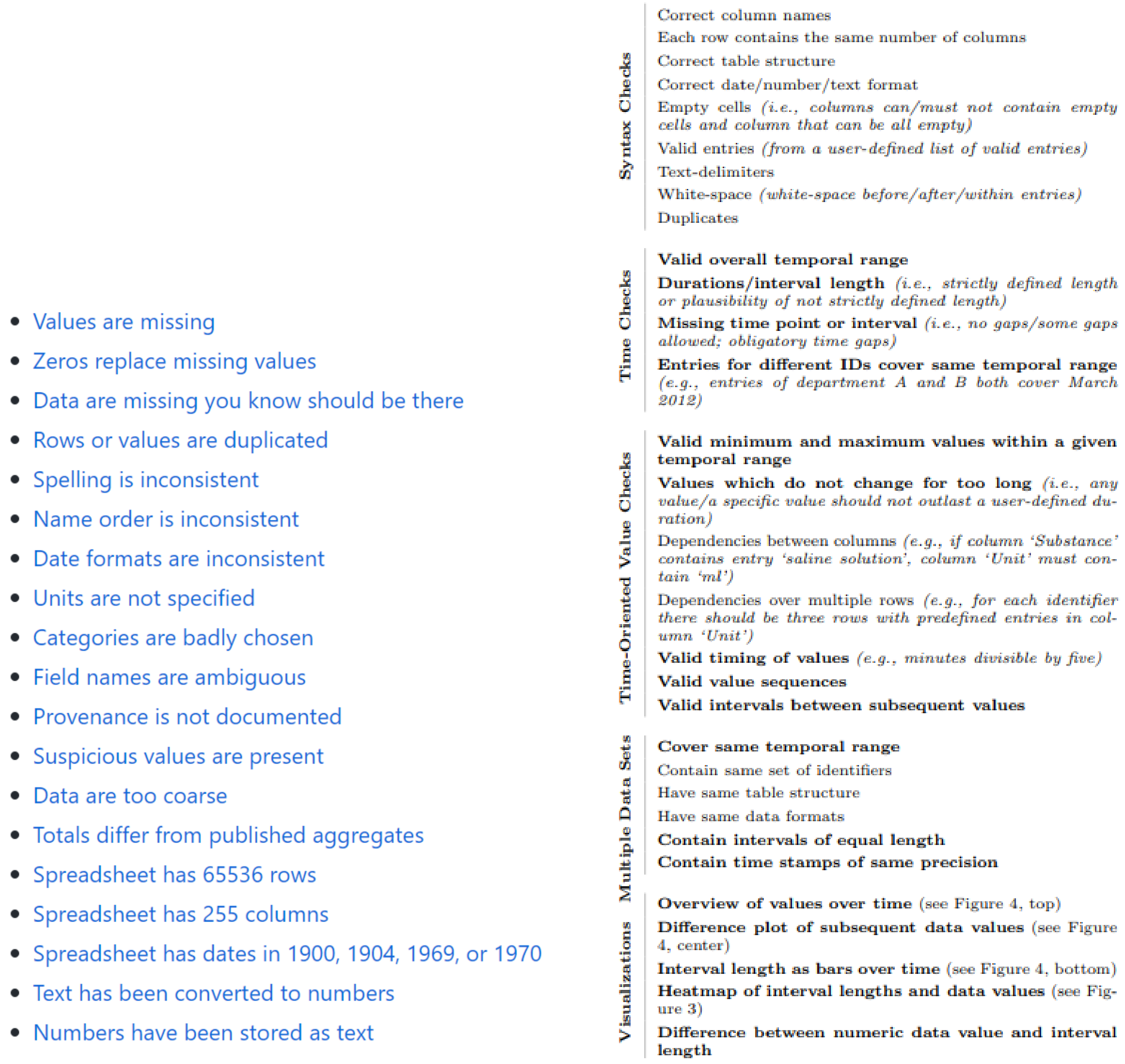
Figure 7: Data obtained from a larger outcome range by varying the process, the red square shows the data from previous figure.

A practical tool for the design of experiments is Browniebee (developed by the Alexandra Institute and some other partners), which can assist in determining the necessary input parameters to explore the outcome space.

3.7 Awareness lists for data inspection

Depending on your organization and application, different types of errors occur more frequently than others. For instance, in some systems, the locations of the IoT devices are set manually in an app, whereas in others they are automatically sent from a GPS. In the first case, human input errors should be monitored. In the second case, GPS fix errors are more typical. Hence, the process for correcting the errors should be different.

Therefore, it is best practice to share lists of typical errors seen in the particular application. As inspiration to awareness lists, the lists are recommended as inspiration from “Bad data guide” which contains typical errors seen in datasets. As a quality inspection guide “TimeCleanser: A Visual Analytics Approach for Data Cleansing of Time-Oriented Data” is recommended.



- Values are missing
- Zeros replace missing values
- Data are missing you know should be there
- Rows or values are duplicated
- Spelling is inconsistent
- Name order is inconsistent
- Date formats are inconsistent
- Units are not specified
- Categories are badly chosen
- Field names are ambiguous
- Provenance is not documented
- Suspicious values are present
- Data are too coarse
- Totals differ from published aggregates
- Spreadsheet has 65536 rows
- Spreadsheet has 255 columns
- Spreadsheet has dates in 1900, 1904, 1969, or 1970
- Text has been converted to numbers
- Numbers have been stored as text

Syntax Checks	Correct column names
	Each row contains the same number of columns
	Correct table structure
	Correct date/number/text format
	Empty cells *(i.e., columns can/must not contain empty cells and column that can be all empty)*
	Valid entries *(from a user-defined list of valid entries)*
	Text-delimiters
	White-space *(white-space before/after/within entries)*
	Duplicates
Time Checks	**Valid overall temporal range**
	Durations/interval length *(i.e., strictly defined length or plausibility of not strictly defined length)*
	Missing time point or interval *(i.e., no gaps/some gaps allowed; obligatory time gaps)*
	Entries for different IDs cover same temporal range *(e.g., entries of department A and B both cover March 2012)*
Time-Oriented Value Checks	**Valid minimum and maximum values within a given temporal range**
	Values which do not change for too long *(i.e., any value/a specific value should not outlast a user-defined duration)*
	Dependencies between columns *(e.g., if column ‘Substance’ contains entry ‘saline solution’, column ‘Unit’ must contain ‘ml’)*
	Dependencies over multiple rows *(e.g., for each identifier there should be three rows with predefined entries in column ‘Unit’)*
	Valid timing of values *(e.g., minutes divisible by five)*
	Valid value sequences
	Valid intervals between subsequent values
Multiple Data Sets	**Cover same temporal range**
	Contain same set of identifiers
	Have same table structure
	Have same data formats
	Contain intervals of equal length
	Contain time stamps of same precision
Visualizations	**Overview of values over time** (see Figure 4, top)
	Difference plot of subsequent data values (see Figure 4, center)
	Interval length as bars over time (see Figure 4, bottom)
	Heatmap of interval lengths and data values (see Figure 3)
	Difference between numeric data value and interval length

Figure 8: Left: Typical errors found in data from Bad data guide. Right: Data inspection best practice list.

3.8 Treat data errors as friends not foes

In many cases there is valuable information to be found in the data errors in the system. The errors in the data often disclose errors in the design process, knowledge found in the organization, processes that are used in relation to the data or many other aspects of the organization and operation. Hence, it is important not just to throw away the errors found in the data validation process, but also to consider and analyze how they can be used for improvement.

The most notable example of this is spam filters in email. If the emails filtered out by algorithms and flagged by users, the resources needed to create new spam filter rules would be much higher than today. So before throwing away the errors in your data – think of placing the data in a separate database.

4 IoT data validation process

The IoT data validation process is a combination of the more general data validation process described in the *Data Validation Process Life Cycle* from the paper “Methodology for data validation 1.0” [Marco Di Zio et al., 2016] and the practical best practices we have learnt from our discussions with other practitioners. It is shown in Figure 9 as consisting of seven steps. In the following section the individual steps in the process are described.

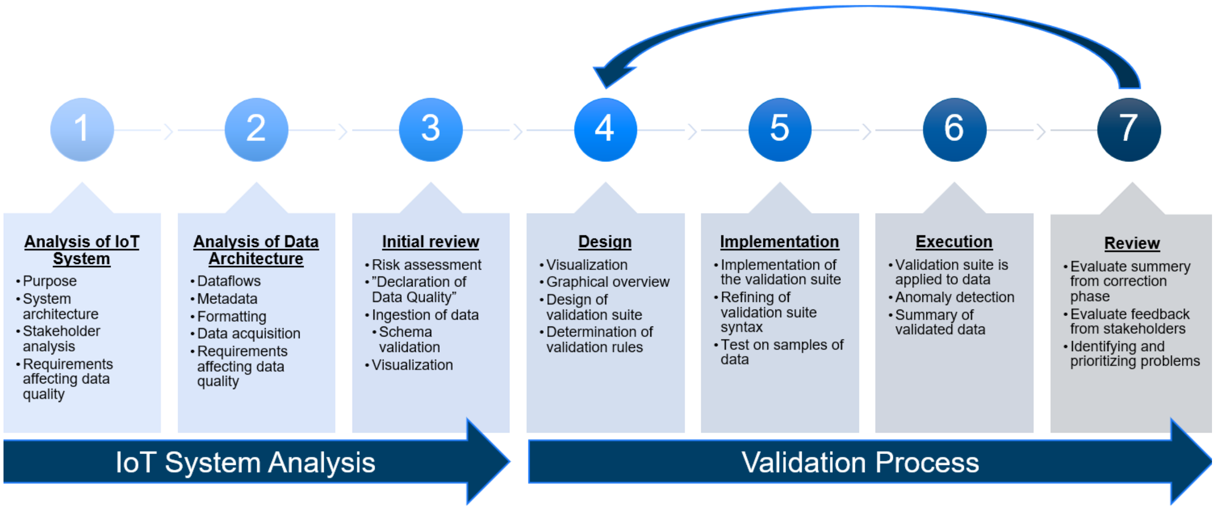

1
2
3
4
5
6
7
Analysis of IoT System
• Purpose
• System architecture
• Stakeholder analysis
• Requirements affecting data quality
Analysis of Data Architecture
• Dataflows
• Metadata
• Formatting
• Data acquisition
• Requirements affecting data quality
Initial review
• Risk assessment
• ”Declaration of Data Quality”
• Ingestion of data
• Schema validation
• Visualization
Design
• Visualization
• Graphical overview
• Design of validation suite
• Determination of validation rules
Implementation
• Implementation of the validation suite
• Refining of validation suite syntax
• Test on samples of data
Execution
• Validation suite is applied to data
• Anomaly detection
• Summary of validated data
Review
• Evaluate summery from correction phase
• Evaluate feedback from stakeholders
• Identifying and prioritizing problems
IoT System Analysis
Validation Process

Figure 9: IoT data validation process.

4.1 IoT System Analysis

The IoT system analysis is the first part of the data validation process and consists of three steps. The aim of these steps is to analyze the system in its entirety, to gain insight into the subsystems and the data architecture investigating how each part might affect the data quality.

4.1.1 Analysis of IoT system

In the analysis of the IoT system, the entire system is to be broken down and explained in detail. This is to ensure that all internal and external parties have a thorough understanding of the system.

Purpose
The purpose of having the IoT system is described, i.e., why the data is gathered and what it is going to be used for, as well as the purpose of the data validation, i.e., what is the reason behind doing the data validation.

System architecture
As part of the system analysis, the physical entities of the system are to be described in detail. What entities

the system consists of and how they are interconnected. Thorough descriptions in this section will allow the entire team to understand the system, which will make the validation process and general communication much easier.

Stakeholder analysis
Any system will have a list of stakeholders that should be specified. The stakeholders of a system will range from the project initiator and project owner to the suppliers of the hardware and software. It is essential to dedicate time for consideration of who would be affected by the system or otherwise have impact on that system. There might be details that has been overlooked, which will ultimately cause a reduction of performance and data quality.

Requirements affecting the data quality
Analyzing what requirements applies to the quality of the data should be handled on a subsystem-specific level and a high level, from the regulatory regulations that applies to the system, or the use of the system, through requirements for the sensitivity of the sensors. The analysis of all the requirements will aid the understanding of the quality of data that is obtainable.

4.1.2 Analysis of Data Architecture

The next step is the analysis of the data architecture, which is in part an analysis, and in part a decisive process.

Dataflows
It is advisable to begin the analysis of the data architecture by creating a model that specifies the data path from phenomena that are being monitored. In other words, what the data will show to the end-user of the data, what storage facilities will the data pass, whether other data is being added from external sources, how the data is accessed throughout, and any other details that is part of the flow.

Metadata
The metadata (data about the data) is an important aspect, that should be decided upon. The decisions should begin by analyzing what metadata is available and possibly already included, and should end with a set of guidelines, describing what metadata is needed in the dataset for the data to meet the expectations from the declaration of data quality, which is described further down.

Formatting
When knowing what metadata to include, the structure of the data can be decided. This decision will be part of the initial review, which will determine whether the actual structure of the data follows the guidelines decided upon.

Data acquisition
Last part to consider is how the data will be transferred to the external consultants for data validation process.

4.1.3 Initial Review

In the initial review, a first glance is cast on the data to be validated, and this is done through a few tasks.

Risk assessment
A risk assessment is a common tool used in most projects. The risk assessment will list everything that will potentially harm the quality of the data, ranking some different aspects of the risk, and by doing this, ranking the risks depending on how much they should be considered, and possibly mitigated.

"Data Quality Declaration"
When the system has been analyzed on a high-level-, subsystem-, and data acquisition basis, before the validation begins, the data quality should be declared, based on the status of the data quality before any validation is done. Hence, the need for a thorough understanding of the system.

If the data quality of the current system is unknown, or for some other reason cannot be declared, a hunch will suffice. It is essential to state the needed data quality to fulfil the purpose of the system. This will be the foundation for the validation suite, as there must be a marker, to measure the quality against. Otherwise, it cannot be known how far validation process has come.

Ingestion of data (Schema validation)
The first initial validation is to ingest the data and thereby do a schema validation. The purpose of schema validation is to check whether the data is structured uniformly, and that the dataset is complete, in accordance with the decided upon schema definition. When working with IoT data, the datasets can be so massive that it is a difficult task to do manually. For that reason, the formal parsing, validation, and ingestion into a database will help, as the ingestion process can report whether all data live up to the definition.

Visualization
After the data has been ingested into a suiting database, and the initial validation of the structure has been made, the data can be visualized, which will add a second layer to the initial validation. Visualizing the data allows for a quick determination of issues and will lead to the validation process.

4.2 Validation Process

The validation process itself is the second part of the data validation process and consists of four steps. The aim of this part of the process is to do the actual validation after having analyzed the system in the first part.

4.2.1 Design

The initial step is to design the validation suite, which essentially means, determining what activities to perform. After the system has been analyzed and the data has been ingested into the database, the validation process – to improve the quality of the data – is initiated. This process comprises four steps, much like the “Data Validation Process Life Cycle” mentioned above.

Graphical overview
As described above, the initial step is to get an overview of the data, to find possible issues that need correction. Thereby, a graphical overview will aid determination of typical features associated with an IoT system:

- Packet size vs. time
- Number of total packages
- Number of devices
- Total volume of data flowing into the system

But most importantly, visualizing the data will make it easy to determine typical errors:

- Missing data
- Discontinuities
- Noise
- Outliers, such as large values

Finding errors is a first step in the design of a validation suite.

Design of validation suite
The design of the validation suite is essentially planning the necessary activities, that is needed to “clean up” the data by analyzing errors and determining how to mitigate them and prioritize the order of the activities.

The suite thereby includes activities to perform, processes to go through, and algorithms/software to use.

When designing the validation suite, it should be considered that the validation process is an iterative process, and that the activities decided upon will be revisited, and it is thereby not necessarily important to correct all errors in the first run.

A validation suite design could include:

- Review of IoT system purpose. Does it need updating?
- Review of the data validation purpose. Should it be redefined?
- Considerations for the dimensions from the Data Quality Declaration, with specific activities to be performed to improve the quality of each dimension in order to meet the requirements.
- Considerations for the risks listed in the risk analysis, and activities to be performed, to mitigate the risks.
- An anomaly list from the visualization and graphical overview, which will be labeled with the affected dimensions, and an activity to correct or mitigate the anomaly.

When the design suite is finalized, the activities should be prioritized, considering the validation purpose and the purpose of having the IoT system. Furthermore, the importance of an activity, and the resources needed, should be considered when prioritizing the activities.

Determination of validation rules
The validation rules should be determined by comparing the activity plan and the "Data Quality Declaration". The validation rules are a set of requirements or guidelines, that the data should abide by, to live up to the desired quality, stated in the data quality declaration. These validation rules will be used in the execution and review phase, as a baseline for analyzing the quality of the data that has been processes with the validation suite.

4.2.2 Implementation

Implementation of the validation suite
The implementation of the validation suite, not to be confused with its execution, is the phase where the validation suite and rules are refined, and the validation suite is tested on sample data to see how data will be affected to make sure that the validation suite performs as expected.

Refining of validation suite syntax
When designing the validation suite, many different tools are used, and many notes are taken, and this might not be structured sufficiently. The first task of the implementation is thereby to structure the validation suite and validation rules in common syntax, making it understandable for everyone on the team as well as the stakeholders.

Test on sample data
Before applying the suite on the real data, the validation suite is applied on some sample data, that resembles the real data, to make sure that the validation suite handles the data as intended and ensure that the data will not be corrupted.

4.2.3 Execution

Validation suite is applied to data
When it has been tested and approved, that the validation suite will not negatively impact the data, in the implementation phase, the validation suite is applied to the real data.

Anomaly detection
Having applied the data validation suite to the data, which should take care of missing data, outliers, and other anomalies, the data should be visualized and analyzed again, to see what effect the validation suite has had, on the real data.

Does anything seem off?
Typical anomalies will be:

- Outliers
- Contextual (Does the value fit the context)
- Gaps

But anything that might seems off should be noted and included in the summary.

Summary of validated data
Last part of the execution is to report what activities have been performed, what effects it has had on the real data, and what might not have been addressed. Just as for the refinement of the validation suite syntax, the summary should be made with a common syntax for all to understand.

4.2.4 Review

Evaluate summary from execution phase, and feedback from stakeholders
Having gone through steps four to six, the validation suite has been designed and applied to the real data, and the result has been summarized. This summary is discussed and evaluated.

The stakeholders should provide feedback during the evaluation phase. Comparing the implementation summary and stakeholder feedback will help to understand what the status of the data is, considering the validation rules, which has been formulated with the declaration of data quality as a basis.

Identifying and prioritizing problems
The evaluation will be the foundation for a reflection on whether the validation suite should be redesigned, only needs minor tweaking, or if it has successfully validated the data, and is thereby a robust solution. Just as in the design phase, the problems that might persist, should be identified, and yet another activity plan can be completed. This activity plan, as well as the evaluation is brought back into the iteration, starting at step four, with the design of the validation suite.

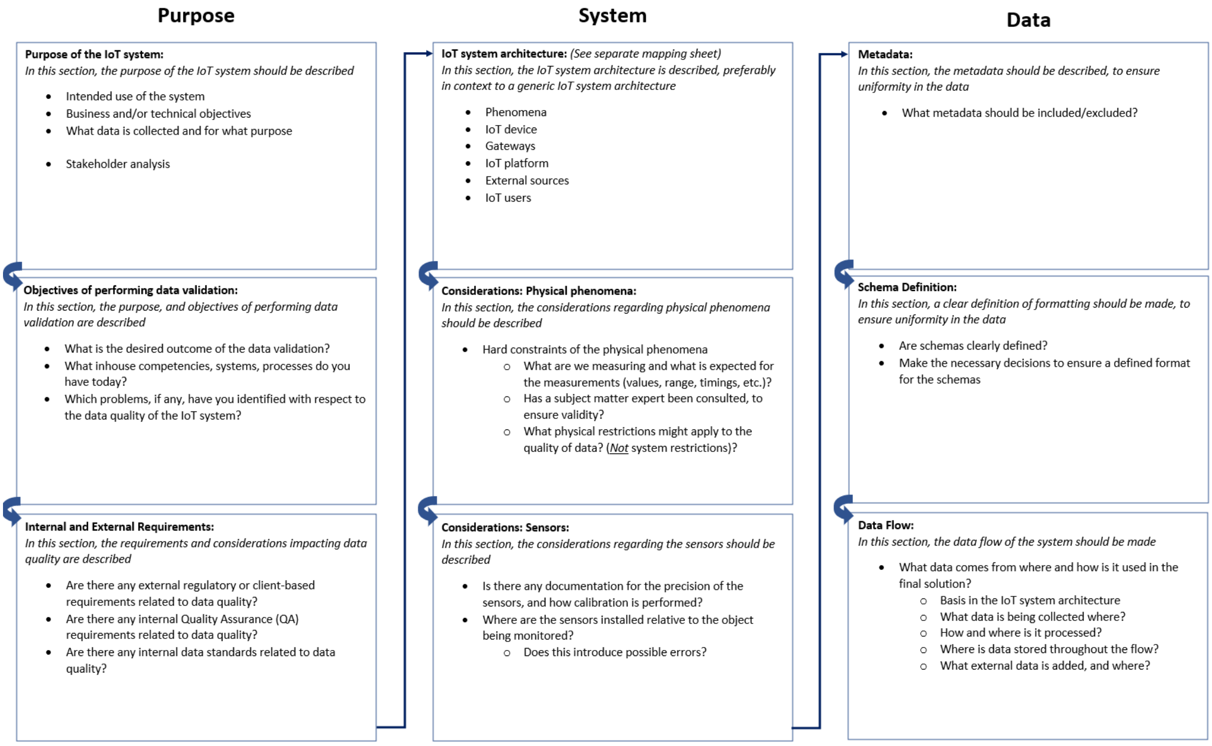
5 Tools for Structuring IoT Data Validation

To guide the process of IoT System Analysis and Data Validation process, FORCE Technology has designed tools that, when used, will ensure a thorough run-through.

5.1 System Analysis Canvas

The canvas tool seen in Figure 10 is the first tool to use, which will guide the analysis of the two first steps: “Analysis of IoT System” and “Analysis of Data Architecture”.

It has been divided into three main sections, that each has three sub-sections, that all comprise questions that will help analyze the system. The task is to fill in each of the subsections, as they go into detail with different aspects of the system as a whole. It should be kept in mind, that it is not strictly necessary to stick to the questions presented in the canvas, as these are merely a guide and inspiration.



Purpose

Purpose of the IoT system:
In this section, the purpose of the IoT system should be described

- Intended use of the system
- Business and/or technical objectives
- What data is collected and for what purpose
- Stakeholder analysis

Objectives of performing data validation:
In this section, the purpose, and objectives of performing data validation are described

- What is the desired outcome of the data validation?
- What inhouse competencies, systems, processes do you have today?
- Which problems, if any, have you identified with respect to the data quality of the IoT system?

Internal and External Requirements:
In this section, the requirements and considerations impacting data quality are described

- Are there any external regulatory or client-based requirements related to data quality?
- Are there any internal Quality Assurance (QA) requirements related to data quality?
- Are there any internal data standards related to data quality?

System

IoT system architecture: *(See separate mapping sheet)*
In this section, the IoT system architecture is described, preferably in context to a generic IoT system architecture

- Phenomena
- IoT device
- Gateways
- IoT platform
- External sources
- IoT users

Considerations: Physical phenomena:
In this section, the considerations regarding physical phenomena should be described

- Hard constraints of the physical phenomena
  - What are we measuring and what is expected for the measurements (values, range, timings, etc.)?
  - Has a subject matter expert been consulted, to ensure validity?
  - What physical restrictions might apply to the quality of data? (Not system restrictions)?

Considerations: Sensors:
In this section, the considerations regarding the sensors should be described

- Is there any documentation for the precision of the sensors, and how calibration is performed?
- Where are the sensors installed relative to the object being monitored?
  - Does this introduce possible errors?

Data

Metadata:
In this section, the metadata should be described, to ensure uniformity in the data

- What metadata should be included/excluded?

Schema Definition:
In this section, a clear definition of formatting should be made, to ensure uniformity in the data

- Are schemas clearly defined?
- Make the necessary decisions to ensure a defined format for the schemas

Data Flow:
In this section, the data flow of the system should be made

- What data comes from where and how is it used in the final solution?
  - Basis in the IoT system architecture
  - What data is being collected where?
  - How and where is it processed?
  - Where is data stored throughout the flow?
  - What external data is added, and where?

Figure 10: FORCE Technology Canvas Tool.

5.1.1 Purpose

The first section details why the data is being gathered, and why the data needs validation. Furthermore, as a part of the “purpose of the IoT system”, it is necessary to analyze the stakeholders of the system, since some of the stakeholders should be consulted during the process of the validation in the review phase. Lastly, it should be noted whether the system or the data will be affected by any internal or external regulatory or quality assurance (QA) factors.

5.1.2 System

The second section is the system architecture analysis, which contains a system analysis and two specific consideration tasks to go through. The system architecture will be done by filling in the blocks in Figure 11. Although it will not apply to all IoT systems, many systems can generally be divided into these blocks.

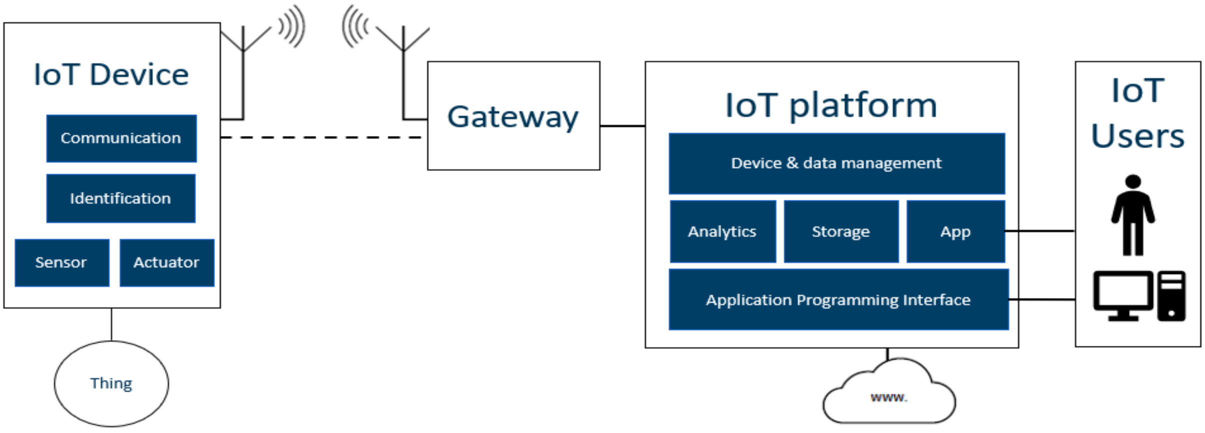


Figure 11: Generalized IoT System architecture.

The task for the analysis is to fill in what hardware/software is used in each block. It will ultimately lead to an entire map of the IoT system.

Upon finalizing the system architecture analysis, an in-depth analysis and mapping of the phenomena, that the sensors are measuring, as well as the sensors themselves are performed. This analysis is part of formulating the requirements to the data.

5.1.3 Data Architecture Analysis

The last section of the canvas is about the data architecture.

Metadata
To ensure consistency in the data, the metadata of the datasets is analyzed regarding what metadata is included on the non-validated data, and decisions are then made about what metadata should be included. Having a clear definition of what metadata to include will ease the process of making a schema definition and thereby a schema validation, which is described in the next chapters.

Schema definition
A schema is a definition of the structure of a dataset. It is the formal definition of the formatting, the division of the data, and for some databases, the relation between various fields and tables.

In our case, formatting refers to the structure of the datasets, e.g., what units to use, how many decimals for numbers, the bit depth of the data, etc. If the schema is poorly defined for the data and the metadata, or if the datasets do not abide by the definition, this will affect the quality of the data. Thereby, a thorough definition of the schema is important.

The task of defining the schema is a decision task, where the exact formatting and schema should be decided upon.

Data flow
The last subsection is the data flow description. It can be done graphically or explanatory and should map out how data is handled throughout the system, from measured phenomena to the end user.

- What data is gathered through the sensors, and how is the data transmitted to the backend?
- What data is applied to the metadata before it reaches the backend?
- What metadata is applied as it reaches the backend?
- How is the data stored throughout the data flow?
- What external data is added to the database, and how is this connected to the measured phenomena?

The data flow should be thoroughly described, as this will, just as the system architecture analysis, enlighten the entire team and make the validation process much easier.

5.1.4 Remarks on the system analysis

What the system analysis does, is to ask the questions “why do we need this validation?” and “what is the data needed for?”, which will aid the determination of when the data quality is fit-for-purpose.

Next it asks, “what is physically possible to measure?” and “what kind of measurements should be expected?”, taking any physical restrictions and requirements into considerations, which will determine the possible quality of the raw data.

5.2 Risk Analysis

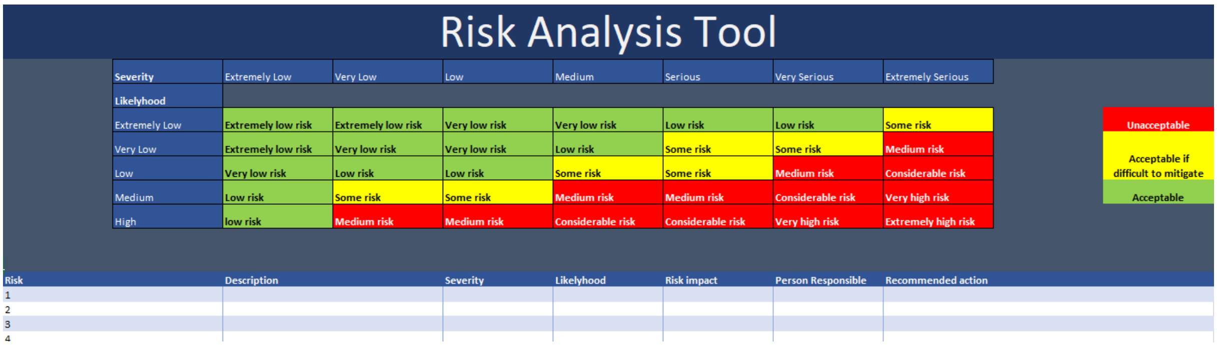
After having performed the IoT system analysis canvas, the risk analysis is a common project management tool, where any known risks to the project are described, with their respective severity, and how they should be handled.

The FORCE Technology-developed risk analysis tool, seen below, presents a table that shows the cross correlation between a risk's likelihood and severity regarding damage to the project. It is important to remember that this risk analysis should be performed with data validation in mind. I.e., what is the likelihood that a particular fault will occur in the system, and what are the business and strategic consequences if this occurs. In the tool the risks are marked with:

- Type
- Description
- The severity to the project, if the risk was to occur
- The likelihood of the risk occurring
- Who is responsible for handling the risk?
- What action is recommended?

The colours of the correlations can be reconsidered, i.e., a medium-severity/very-low-likelihood risk could be yellow instead of green, but it is recommended to use the following:

- **Red**: An unacceptable risk, that needs to be eliminated
- **Yellow**: An acceptable risk if it is mitigated (reduced)
- **Green**: An acceptable risk, that is known but does not need handling



Risk Analysis Tool

Severity	Extremely Low	Very Low	Low	Medium	Serious	Very Serious	Extremely Serious
Likelyhood							
Extremely Low	Extremely low risk	Extremely low risk	Very low risk	Very low risk	Low risk	Low risk	Some risk
Very Low	Extremely low risk	Very low risk	Very low risk	Low risk	Some risk	Some risk	Medium risk
Low	Very low risk	Low risk	Low risk	Some risk	Some risk	Medium risk	Considerable risk
Medium	Low risk	Some risk	Some risk	Medium risk	Medium risk	Considerable risk	Very high risk
High	low risk	Medium risk	Medium risk	Considerable risk	Considerable risk	Very high risk	Extremely high risk

Unacceptable
Acceptable if difficult to mitigate
Acceptable

Risk	Description	Severity	Likelyhood	Risk impact	Person Responsible	Recommended action
1						
2						
3						
4						

Figure 12: Example of risk analysis table for data validation.

It is furthermore recommended to do either two risk analyses or make a system within the same risk analysis that can handle two types of risks.

1. Risks affecting the project or company if the data quality is low
2. Risks that might affect the quality of the data

For example: Risks that affect the project could be marked as "Axe"-risks, and risks to the data quality could be marked "Box"-risks.

Risk	Description	Severity	Likelihood	Risk impact	Responsible	Action
A1	Bad data would mean people get flooded	Very Serious	Low	Medium risk (RED)	Project owner	Perform Data validation Process
B1	Sensor Placement is bad	Medium	Low	Some Risk (Yellow)	Hardware installer	Double check placement of all sensors

Table 2: Example of data validation risk analysis.

5.3 Data Quality Declaration

Based on the data fitness (whether the data is fit-for-purpose), described through the system analysis, the quality of the data should be declared.

First, the data quality should be declared according to the quality of data delivered by the system at the moment of the system analysis. Second, the desired quality of data should be declared.

The data quality declaration is divided into four dimensions:

- Completeness
- Correctness
- Actuality
- Reusability

The data quality declaration used is originally based on the eleven dimensions of quality, mentioned in the learning material of "Data Management Body of Knowledge" [Deborah Henderson et al., 2017].

Accuracy, completeness, consistency, currency, data integrity, precision, privacy, reasonableness, timeliness, uniqueness, validity, and accessibility. These eleven dimensions are consolidated to four dimensions[1], that are used for our data quality declaration. Table 3 shows what each dimension considers.

Completeness	Indicates the degree to which the dataset comprises the expected data elements, regarding the data specifications, and has two aspects: **Entirety** is completeness as to whether the dataset comprises the expected entities (that all entities – being concrete or abstract – are *registered*) **Coverage** is completeness as to whether the dataset comprises the expected information about the registered entities *properties (In a relational database that the necessary attributes have values)*
Correctness	Indicates the degree to which the data values comply with actual values, with two aspects: **Semantic** correctness is if there is compliance between the registered values and the actual physical values **Syntax** correctness is if there is compliance regarding rules of syntax, i.e. rules for spelling and structure
Actuality	Indicates the degree to which the data is timely representative relative to the reality they are representing (is the data *outdated)* *(E.g., if real time data is needed, 10 minutes delay might not be acceptable, but to model and monitor the movement of ice, 6 months old data would suffice)*
Reusability	Indicates the degree to which the data is understandable, and can be used by others, and has two aspects: **Understandability** is whether the data is represented in a way that is easily readable for the users (uniform, readability, documented by *metadata*) **Consistency** is whether the data has contradictions and is coherent to other data (unmistakable, coherence between data in the dataset, coherence with specification)

Table 3: The four dimensions of quality.

[1] cf. "Fælles sprog for datakvalitet" [Digitaliseringsstyrelsen, 2019, in Danish]

Each dimension should be marked with a zero to five ranking, expressing whether the data is useful or not, using the following ranking:

☆☆☆☆☆	Unfit for the intended use
★☆☆☆☆	Barely usable for the intended use
★★☆☆☆	Usable, but with many errors/unintentional results
★★★☆☆	Usable, but with errors/unintentional results
★★★★☆	Usable, but with a small number of errors/unintentional results
★★★★★	Can be used with no further considerations

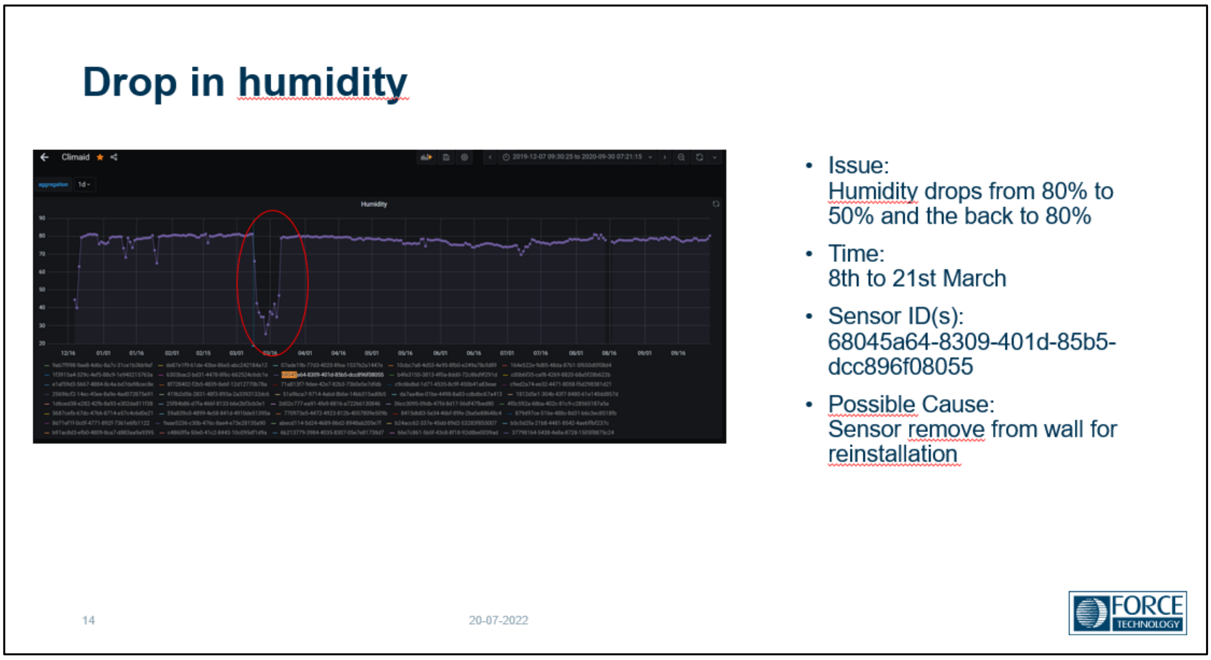
As stated above, it is important to fill in two declarations, as the first will clarify the status of the current system and the second will describe the intended data quality, for the data to be fit-for-purpose.

The difference between the two declarations will aid the design phase of the validation process, as this can reveal some known errors that need corrective actions, and furthermore provide direction for the design of the validation suite and rules.

5.4 Anomaly tracking

In most IoT systems seen, even from large organizations, there are many errors in the data once one starts looking for them. The first thing is to identify the errors and afterwards to do something about them. If you start working with the first error you encounter, you will most probably not find the most critical ones. Therefore, it is a good practice to keep an anomaly tracking sheet when performing data validation. This can be based on any format you like, but at FORCE Technology we work with two different formats: slides and tables. The slides function as a tool to document what has been found. Typically, with a simple screenshot of the error and the identifiers to be able to find it again and quickly determine what was the problem.

Following this, the anomalies can be prioritized and addressed according to their ranking and the available resources.



Drop in humidity

- Issue:
  Humidity drops from 80% to 50% and the back to 80%
- Time:
  8th to 21st March
- Sensor ID(s):
  68045a64-8309-401d-85b5-dcc896f08055
- Possible Cause:
  Sensor remove from wall for reinstallation

14

20-07-2022

Figure 13: Example of slide to track anomalies.

5.5 Putting activities into a routing

As mentioned previously data validation is an iterative process. The challenge with starting data validation is that typically you find that there are not one, but many things that need to be addressed. Therefore, it is essential to prioritize the activities to ensure progress and to only spend resources on the most essential challenges.

It is a simple but effective tool in table format with the columns:

- ID: Identifier to track the activity
- Description: A short description of the activity
- Activity responsible: Who is the lead on the activity
- Business priority: Importance of the result in relation to business needs scored from one to five
- Estimated time duration: The calendar days, weeks, months it takes to do the activity
- Estimated cost to correct: Cost of hours and materials to perform the activity scored from one to five
- Total priority: the multiplication of business priority and estimated cost

Please note that some of these activities are repetitive actions, for driving the overall data validation effort. These could be:

- Perform the quarterly review and update of the data validation process and document
- Weekly visual search for anomalies and update the anomaly tracking sheet
- Annual update of most valuable data and sensors list

In this way the document also serves as the main list of how the iterative data validation process is organized and who is responsible for it.

6 Tools for the Validation Process

Going through the system analysis provides a complete overview of the system in its entirety as well as a risk analysis, two data quality declarations (as-is and to-be), an anomaly tracking document, and an activity prioritization list. Based on this the iterative part of the data validation cycle can begin, where tools are designed and implemented to validate the data. The tools can be broken into five categories:

- Software tool stack
- Domain expert tools
- Visual analysis tools
- Statistical and algorithmic tools
- Calibrations

Throughout the cycle, some data management tools are used. What exact tools to use depend on how the data validation suite is designed, and will differ depending on the exact system, but an example of a validation toolbox is described in the following sections.

6.1 Software tool stack

A common stack of tools common for IoT systems can also be used for the data validation. However, the added value is in how to use these tools. The typical tool stack used at FORCE Technology together with the Alexandra Institute for the initial data validation is the following:

Node-RED

Node-RED is a low-code, visual, flow-based programming interface, that is easy to use and open-source. Note-RED is used for several steps of the validation, from ingesting the data, to performing various operations on the data to correct or mitigate errors.

PostgreSQL

PostgreSQL is a feature-rich open-source database. It can both store structured data (relational, column-based) and semi-structured data such as JSON blobs or even text or binary blobs. The Timescale extension enables efficient handling of time-series, which is an essential capability for IoT data. Finally, the PostGIS extension adds capabilities for geographic data.

Grafana

Grafana is a data visualization app which uses SQL queries to draw data from a database and visualize it in an interactive dashboard. This is used to create a visual overview of the data.

Jupyter Notebook

Jupyter Notebook is an interactive programming environment, mostly for Python, which can be used to run statistics and other analytics on the data. It has a Notebook functionality to it, making it easy to mix notes and programming. Through this, one can have some code that is immediately followed by some explanatory text or similar.

This stack is versatile enough to be deployed in different ways, such as in a cloud or private server, or on the personal computer of the person(s) performing the analysis, or even on-premises or on the edge if the data needs to stay close to the place where it is produced.

6.1.1 Using the stack

As stated above, these tools are merely one example of a stack that can be used for the data validation, and whether to use these or not is to be determined in the design phase. Similar software tools are available, but we contribute with open-source modules mostly for the tool stack above.

For example, these tools could be used during the first data validation in the following steps:

1. The data is provided in some data format, this could either be as a file dump in a CSV format, or as access through a REST API.
2. The data is ingested through Node-RED into Timescale (PostgreSQL), with the possibility to perform a schema validation either prior or after insertion into the database. Should there be errors in the schema and format of the data, Node-RED would be able to find the errors already in this step.
3. Having the data successfully ingested into Timescale, Grafana is able to display the data in a visual dashboard, where outliers, gaps between data packets, or other issues can be identified and prioritized.
4. A validation suite, and a set of validation rules can be designed, for example by performing statistical analysis on the data, using Jupyter Notebook.
5. Node-RED is used to perform the decided operations on the data. First on sample data, as part of the implementation phase, next on the actual data as part of the execution phase.
6. The data can once again be visualized using Grafana, and if needed, the data can be analyzed using Jupyter Notebook once more.
7. The evaluation is reviewed, and the remaining issues are identified and prioritized.
8. A new validation suite and set of validation rules are designed, as the process is iterated.

Finally, there is the challenge that the data being validated is just a copy of the original database of the system. Therefore, the last step in the data validation process should be to design how to integrate the data validation tools into the primary system in operation (the production environment).

6.2 Visual tools

Visual tools are a great aid to support how to investigate data manually. They are typically used to find the anomalies and data challenges for the system investigated. Here, we present a few tools embedded in the software tool stack and additional resources that can be combined with the tool stack to support additional activities.

6.2.1 Dashboard

The most important thing is to be able to have an overview of the system, where the most obvious system wide data challenges are found, and then explore in more detail. For this, a dashboard is a great solution. The initial data should include data for a long period, for instance a year as a first view.

The first thing often investigated is a simple overview of the number of connected devices and the number of packets received in the system. This can give the first insights if, for instance, there are connectivity issues or changes in the installation base. Also, to see if scaling will become an issue in the near future.

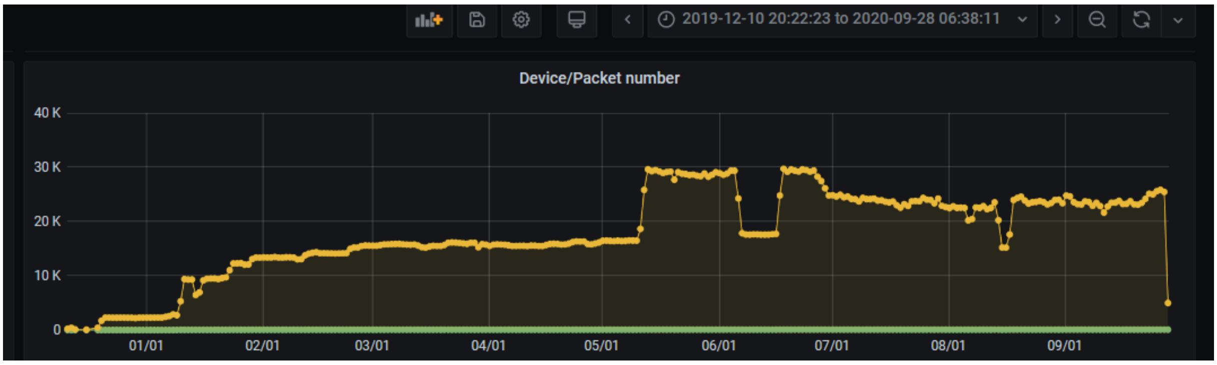


Figure 14: Devices and packet size.

The next tool is to use heatmap scatter plots to visualize the number of packets (colour coding) during a time period (x-axis) between certain sizes (y-axis). Here, changes in formats, payload contents, and time-based abnormalities can quickly be identified.

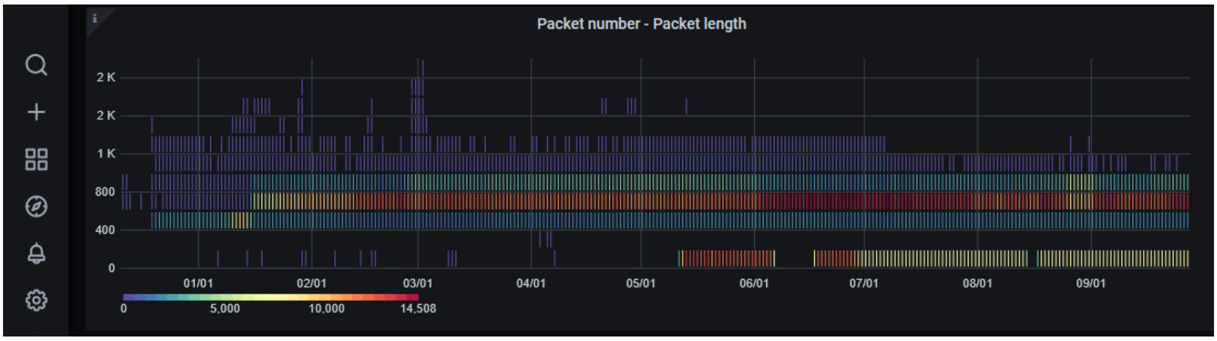
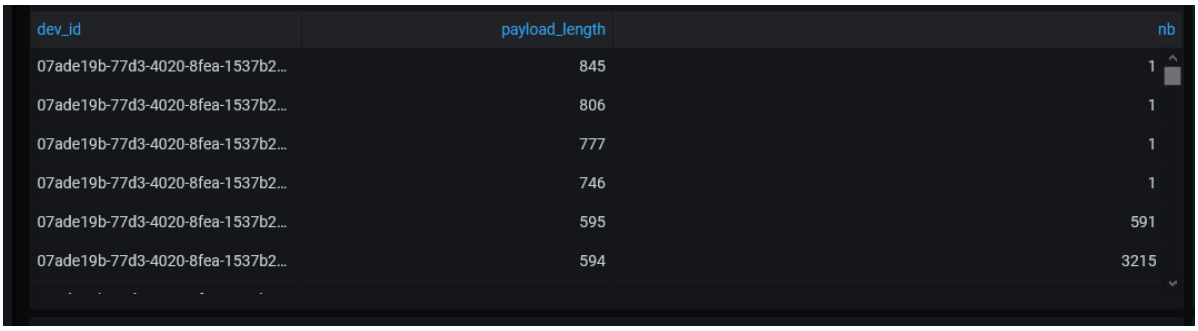


Figure 15: Heatmap binned scatter plot for data visualization.

Tables are also a useful tool in the dashboard, for instance, to show the payload length for individual devices.



dev_id	payload_length	nb
07ade19b-77d3-4020-8fea-1537b2...	845	1
07ade19b-77d3-4020-8fea-1537b2...	806	1
07ade19b-77d3-4020-8fea-1537b2...	777	1
07ade19b-77d3-4020-8fea-1537b2...	746	1
07ade19b-77d3-4020-8fea-1537b2...	595	591
07ade19b-77d3-4020-8fea-1537b2...	594	3215

Figure 16: Table with payload length and number of packets.

It can also be used to show the interarrival times between packets and the number of devices online. Please note that these can be difficult to calculate if there are many sensors, that are event triggered compared to sensors using a scheduled transmission pattern.

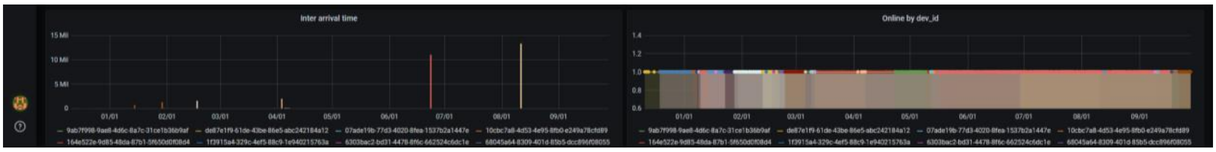


Figure 17: Interarrival times and online indication.

It is also good practice to visualize all data that measures the same parameter in a single plot. Please note that it can be a good idea to segment them into what the sensors are measuring. For instance, in the humidity case in Figure 18 could be segmented into measurement in air, on concrete and in refrigerators.

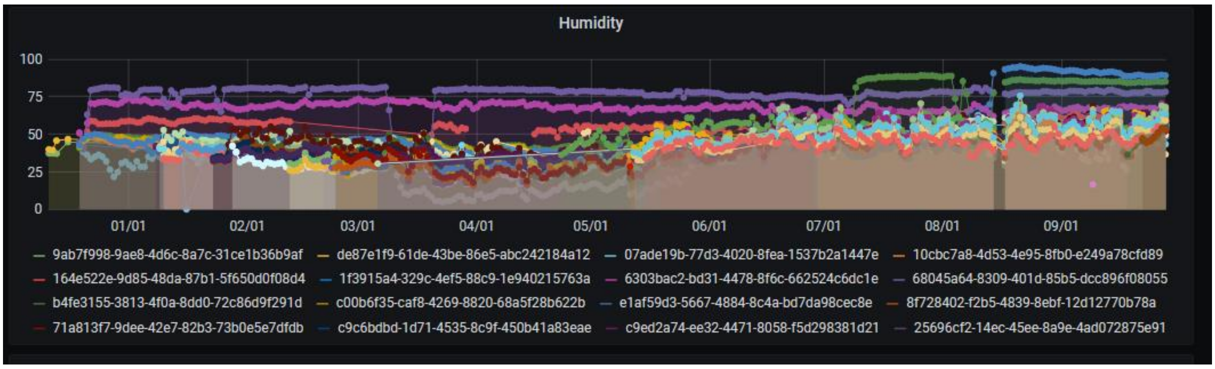


Figure 18: Time series value visualization of all humidity measurements.

In Grafana and many other tools it is a great advantage to explore the individual series by clicking and examining a single variable, looking for particular anomalies. The best strategy is to go from the overview and inspect the individual sources based on what can be observed from the overview, but also to randomly investigate some of the sensors that appears fine. For instance, sensors reporting the same value again and again, can be a source of error to look for. The inspection can reveal many other types of errors such as:

- Missing Data
- Transients and discontinuities which are physically impossible
- Noise in the data
- Outliers, e.g., small or large values

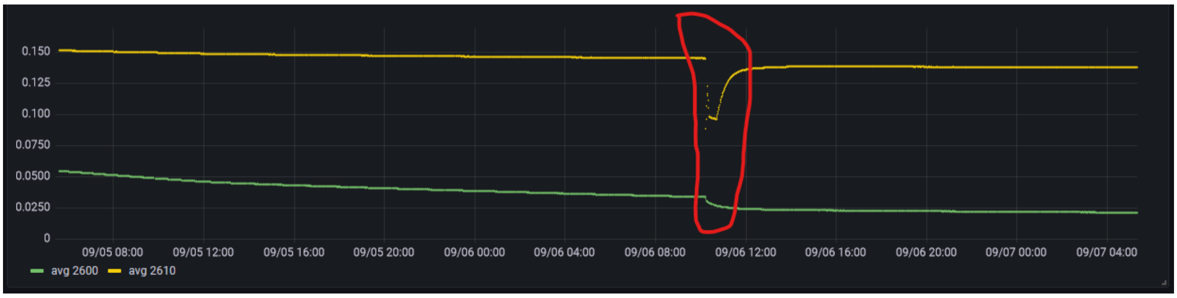


Figure 19: Single data trace visualization - here a discontinuity in a sensor value.

One particular word of caution is to be very aware of time aggregation in the data. In many IoT solutions, the data is measured much more frequently that can be visualized. For instance, if a sensor is measuring data for every minute for a full year, this becomes 525.600 data points. Therefore, data is typically aggregated before visualization to speed up the system. This aggregation is often the average of a time period, such as one day when visualizing a full year. This can hide and obscure missing values and outliers. Hence changing the time aggregation is a good practice. Another way is to change the aggregation functions to include maximum, minimum, standard deviation, in addition to average for the time period.

The list of visualizations of the data can go on and on, and they will depend on the application of the system. The above tools are general and should be found in all IoT systems. A last an important note is geographical visualizations, which can be very helpful in determining if there are location specific data quality issues. This can show the mean values, the missing data percentage or the time variation of data in a specific area.

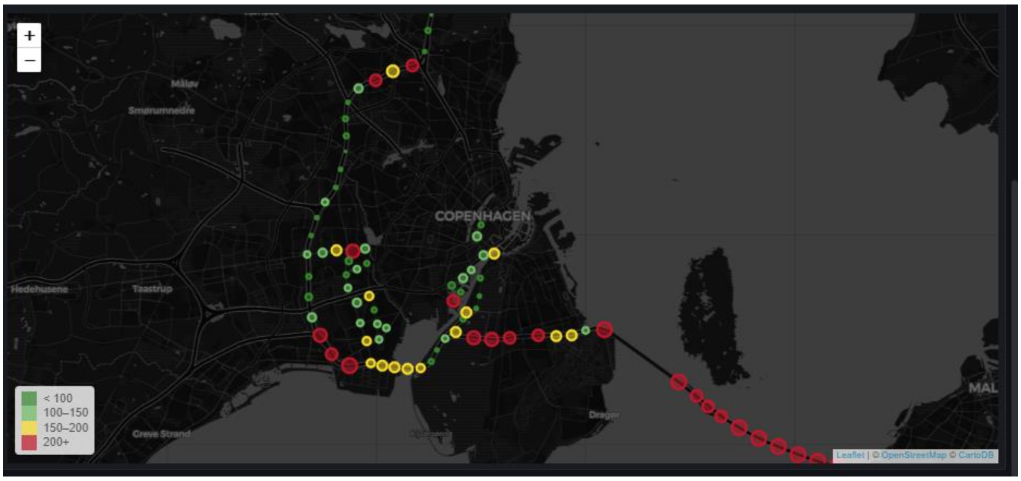


Figure 20: Geographical visualization of data.

6.2.2 Alternative visualizations

Changing visualizations can be a good way to investigate data in a new way. Sometimes changing the visualization can help to see new challenges.

By changing visualization types, new patterns in errors can become apparent and be included in the overview representation. Treevis.net is a site containing a multitude of visualization techniques for tree structures – this can be a great source of inspiration. The link in the top left also contains other types of visualizations.

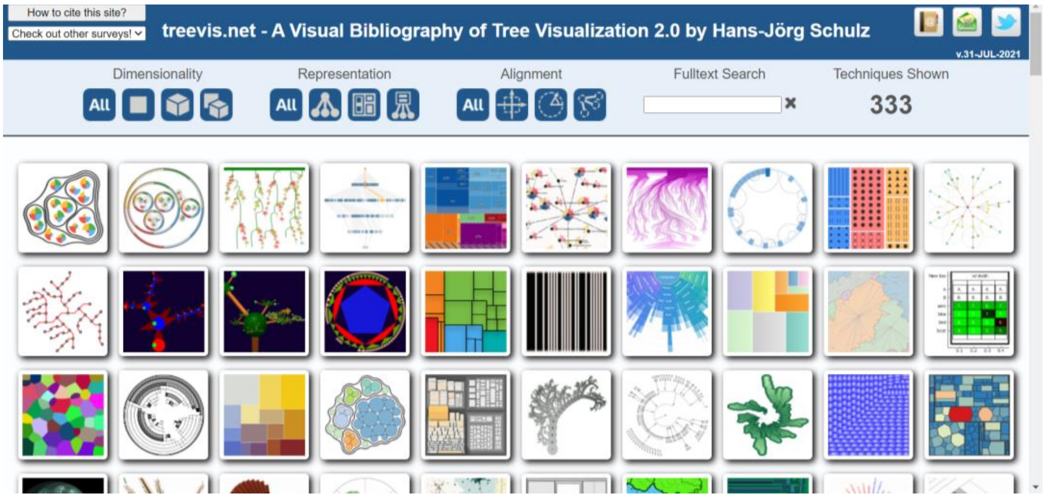
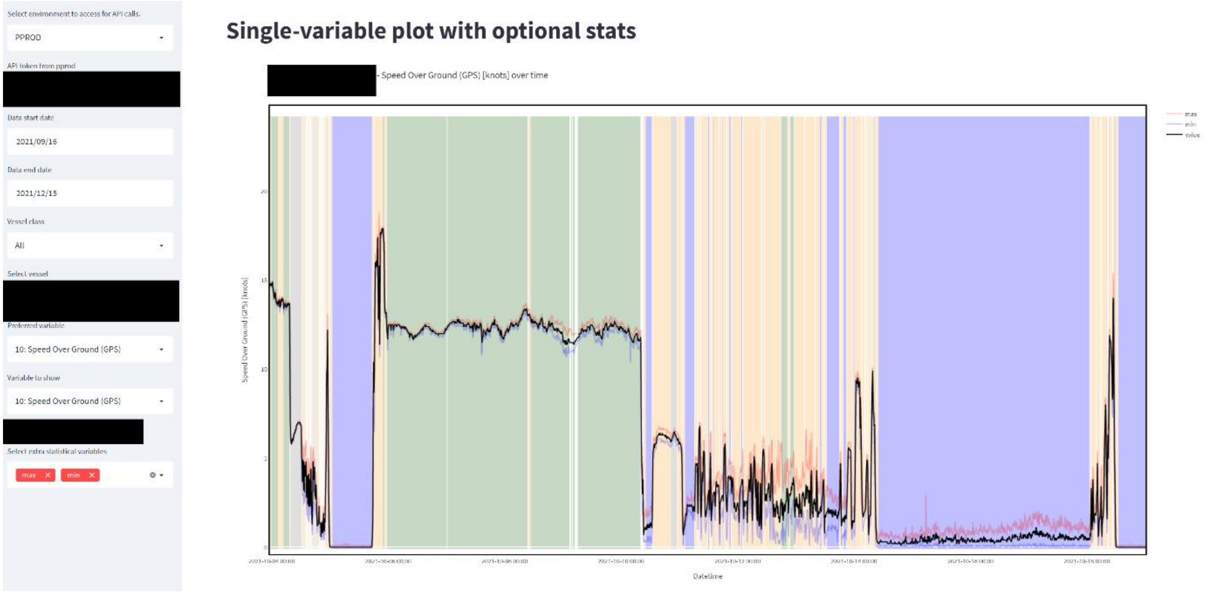


Figure 21: Examples of visualizations of data.

A particular visualization that can be used for single variables – i.e., data from a single sensor is the statistical single variable plot. Figure 22 shows a visualization from the [streamlit package from GitHub](), although the documentation is better at [streamlit.io](). The advantage of this is that plots can be configured rapidly to explore the data. That is, which data source to investigate and what statistical values should be shown.

The background colour indicates the status of the system, in this case if the ship is manoeuvring, steady sailing or at a port. The black is mean, red is maximum, and blue is minimum.



Select environment to access for API calls.
PPROD

API token from pprod
[redacted]

Data start date
2021/09/16

Data end date
2021/12/15

Vessel class
All

Select vessel
[redacted]

Preferred variable
10: Speed Over Ground (GPS)

Variable to show
10: Speed Over Ground (GPS)

[redacted]

Select extra statistical variables
max × min ×

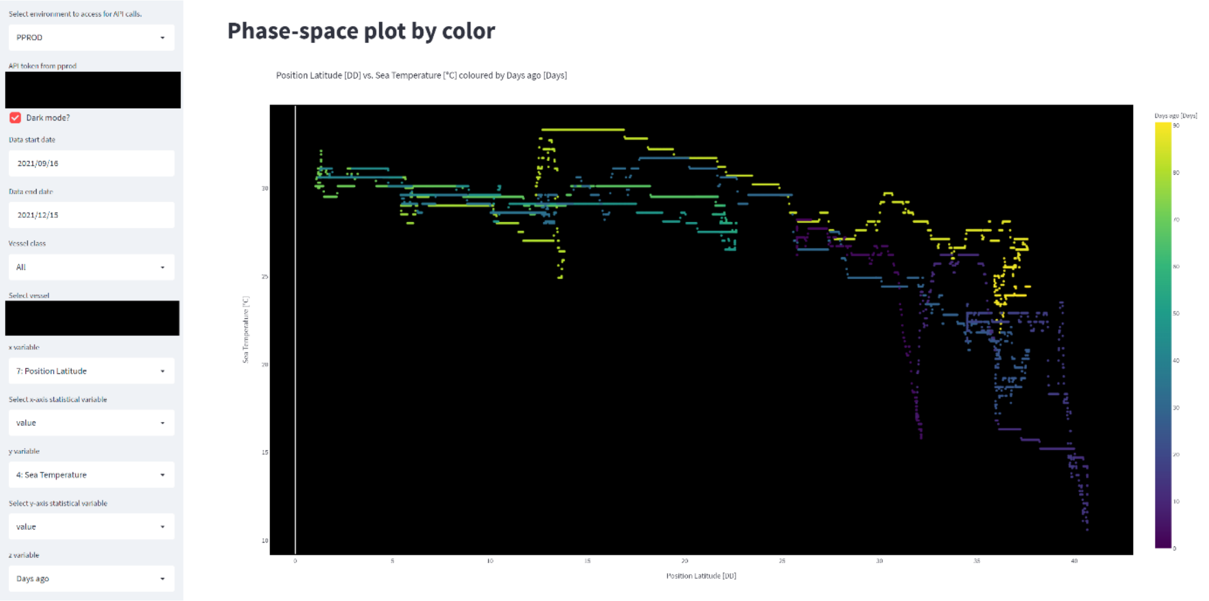
Single-variable plot with optional stats

[redacted] - Speed Over Ground (GPS) [knots] over time

Figure 22: Statistical single variable plot.

Another great visualization is the phase-space plot. Here the user can select the three data sources which should be compared by plotting them as x, y and colour. This gives the opportunity to see if a specific anomaly in the relation between the two variables has happened at a certain location or a given time, since this will be highlighted as a specific colour.

One way to add additional detail into this visualization is to use Scagnostics, which is a method for determining the distribution of a scatter plot in relation to characteristics like outlying, skewed, clumpy, convex, skinny, striated, stringy, straight and monotonic.



Select environment to access for API calls.
PPROD

API token from pprod
[redacted]

☑ Dark mode?

Data start date
2021/09/16

Data end date
2021/12/15

Vessel class
All

Select vessel
[redacted]

x variable
7: Position Latitude

Select x-axis statistical variable
value

y variable
4: Sea Temperature

Select y-axis statistical variable
value

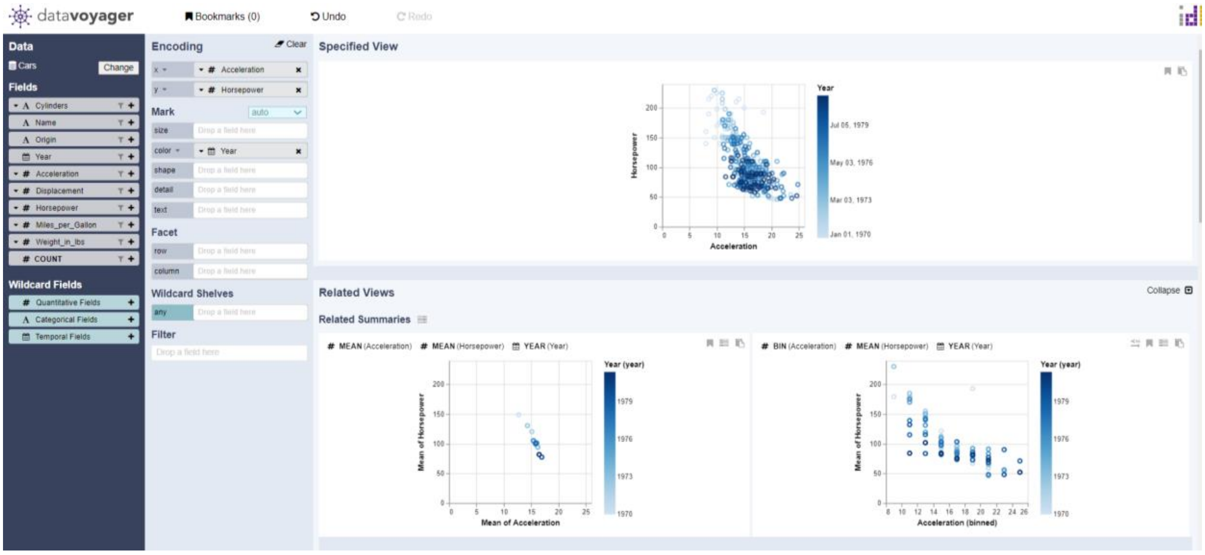
z variable
Days ago

Phase-space plot by color

Position Latitude [DD] vs. Sea Temperature [°C] coloured by Days ago [Days]

Figure 23: Phase space plot.

Similar to the streamlit implementation, the Vega framework for visualization can also be suggested. It provides a uniform way of describing visualizations. In particular, the voyager graphical user interface provides nice features embedded into a web browser similar to the phase-space plot and statistical single variable plot mentioned above.



x: Acceleration
y: Horsepower
Mark: auto
color: Year

Figure 24: Voyager 2 Vega based visualization of data.

A final point about graphical validation is to use the user interface also. Sometimes, the graphical tools for visualizing data for users of the IoT data has built in functions, which alters the data that the users see, compared to the data in the data base. This could for instance be:

- Labelling and units for the data.
- Adjustment of scale of y-axis.
- Correction of offset and gain.
- Conversion from one parameter to a derived parameter, for example air pressure to altitude.
- (Un)wrapping of data from unsigned to signed.
- Time aggregation.

The exercise is quite simply to explore if the data visualization used in the data validation process aligns with the data in the graphical user interface.

6.3 Domain expert tools

One of the key challenges when performing data validation is to know what is normal and what is not normal and this can be defined by domain expert but in order to avoid disturbing the subject matter/domain expert every time data should be validated, some tools come in handy for the data validation responsible.

6.3.1 Understanding the physical system

Understanding the physical system is essential for validating the data. Here physical models are an essential tool. By physical models we do not mean scaled representations, but rather a diagram of the physical model. These can be of very high complexity, but the essence is to create a model, that describes the relationships and constraints between multiple data elements. They should be created in collaboration with a subject matter expert (SME). There is a tendency to overdo these models, the essential is to start simple and build from there. Even a simple description can be very beneficial. As an example, a ship is used:

- What are typical, minimum, and maximum speeds when sailing/manoeuvring?
- How fast can it change direction or accelerate?
- What would a typical fuel consumption, engine speed and oil temperature be?
- What is the difference in the above between vessel class A, B and C?

Having a crude drawing (physical model) of a ship with such parameters noted down, greatly improves the understanding of the data constrains for the data validation process.

In general, it can be formulated as the five following steps:

1. Identify objective of model.
2. Draw a schematic diagram including data variables (and list constraints).
3. Determine physical dependencies.
4. Write dynamic balances and relations (mass, energy, heat, flow, geometry).
5. Verify with comparison to actual data (simulated/measured/observed).

As a simple example, take a system which monitors rainwater in a rainwater reservoir to compensate for heavy rain due to climate change. Domain experts are able to make reasonable estimations of parameters such as rainfall capacity, reservoir in-flow and out-flow, cf. Figure 25.

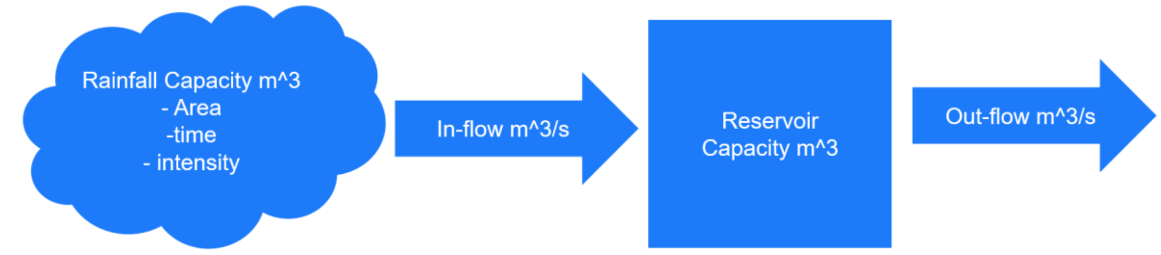


Figure 25: Example of very basic modelling of a physical system being monitored.

If it is found that there is not a satisfactory alignment between the physical model expectations and the data measured from the system, the next step can be to improve the model by using more advanced models, such as system simulations or 3D Multiphysics simulation tools such as:

- Fluid flow and heat transfer
- Structural mechanics
- Chemical processes
- Acoustics
- Electromagnetics

This can be used to determine if sensor placement is correct, if flow estimations from domain experts are correct, and many others. An example could be the placement of heat sensors to measure exhaust temperature from an engine. Here the heat distribution can be simulated in the exhaust to determine the most accurate placement of the sensor.

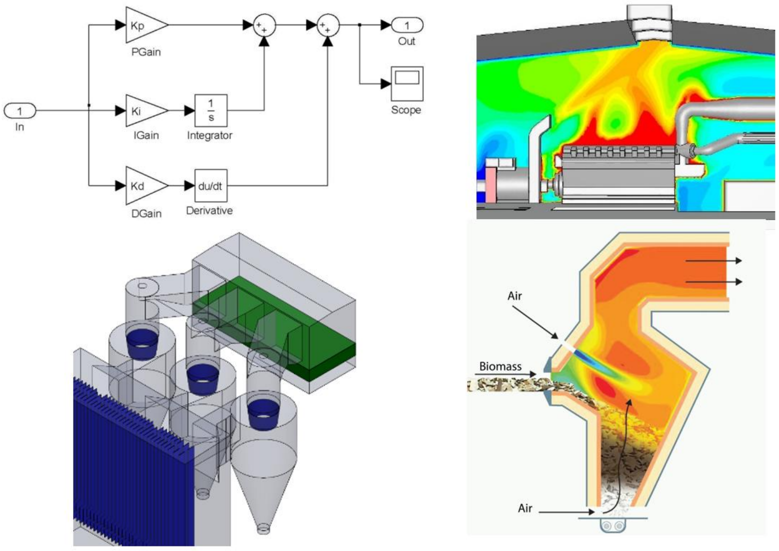


Figure 26: Examples of system and Multiphysics simulations to enhance model accuracy.

As an example of the complexity, the reader is encouraged to see the paper "Non-invasive temperature measurement of turbulent flows of aqueous solutions and gases in pipes" J. Gebhardt, 2020, where measurement of temperature in a pipe is related to the position of the sensor and the uncertainty related to the pipe dimensions.

6.3.2 Installation validation

Installation validation is performed to check the validity of the sensor positions and installation regarding the physical structures it is measuring.

Best described by a simple example of strain gauges mounted on a building (measuring the stress in the concrete), as the activities performed can vary greatly.

- **Position of validation sensors** are inspected to be installed at the location on the building indicated by their metadata (or naming convention).
- **Correctness of sensor installation** determines if the sensors are installed according to their specifications. Such as if a water flow meter is not mounted directly after a 90-degree bend.
- **Physical effect to sensor output validation** ensures that sensors are properly mounted. For instance, if strain gauges are not properly glued to the building, protected by water ingress etc. They will measure strain, but the values will be incorrect. Here the sensors can either be physically inspected, compared to a reference, or dismounted and remounted.
- **Monitoring operating condition** is a special case of this, where related parameters are measured to verify that it does not impact measurements - e.g., monitoring vibrations near a rotation sensor.

It is important to note, that typically these activities will be based on a sampling strategy in such a way, that only some of the sensor installations are validated unless many errors are found.

6.3.3 Comparisons to reference data

In some IoT systems reference data can be available through other publicly available datasets. Often the domain experts are aware of these and can guide in the right direction. This could be:

- Weather data.
- Traffic information (or partial data, e.g., busses).
- Validated metering of electricity, heat, gas, water etc.

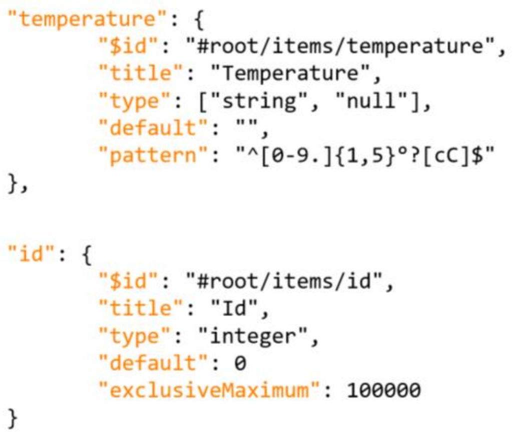
It should be pointed out, that these datasets may, **or may not,** be validated already. Hence it is not guaranteed, that comparison is against a valid data source, but agreement between two independent datasets is a good indicator. The process can either be performed manually or through algorithms. Be aware, that time synchronization of the datasets can give challenges to the algorithmic approach.

6.4 Statistical and algorithmic tools

Statistical and algorithmic tools are essential to automate the data validation, but also in the manual inspection of data. This ranges from simple histogram plots of individual sensors to Bayesian statistics and neural network machine learning models to detect anomalies.

6.4.1 Schema validation

The first approach which should be used is to evaluate the schema of the data. A schema is a definition of the fields in the data being sent, including format and ranges for these formats. For example, a temperature measurement must contain an ID of the device, and this is reported as an integer, the temperature is a string containing only numbers and decimal points. The schema validation process is used to verify that the data coming into the system is compliant. Various schemas already exist, but to the extent possible, it is recommended to use standard ones, and at least using an underlying syntax such as JSON-LD that is favouring interoperability. A good place to start is smartdatamodels.org, but the most important aspect is to have a schema.



```
"temperature": {
        "$id": "#root/items/temperature",
        "title": "Temperature",
        "type": ["string", "null"],
        "default": "",
        "pattern": "^[0-9.]{1,5}°?[cC]$"
},

"id": {
        "$id": "#root/items/id",
        "title": "Id",
        "type": "integer",
        "default": 0
        "exclusiveMaximum": 100000
}
```

Figure 27: Examples of schemas.

Alexandre Institute made an open-source schema validator for Node-RED, node-red-contrib-json-multi-schema.

6.4.2 Constant value detection

A very common error seen in datasets are constant values. This could be as simple as defining a window of ten values where the data from the source is constant. This could indicate that either firmware in a sensor or connectivity is lost and the data values are being replaced by the last known value.

6.4.3 Confidence levels

Confidence level analysis was introduced by “Validating data quality during wet weather monitoring of wastewater treatment plant influents” J. Alferes, et. Al., 2013. Here the concept is to define four levels for each of four parameters for a single data source. The four levels are:

- *Minimum*
  The minimum level that should occur in the data – for instance, the absolute zero temperature of -273 °C
- *Low*
  The lowest value of data – such as -15 °C for the cooling fluid of a car
- *High*
  The highest value of data – such as +95 °C for the cooling fluid of a car
- *Maximum*
  The maximum value that should occur in the data – such as +115 °C for the cooling fluid of a car

Each of these are converted to a confidence level score, such that the confidence below minimum or above maximum is given the value 0, between low and high the score is 100, and between minimum and low, and high and maximum, the value is the linear interpolation between the two values as shown in Figure 28.

The four parameters where this should be evaluated are:

- *Gap filling*
  The number of values missing in the data source over a given time-period.
- *Range Check*
  The actual values in the data.
- *Rate of change*
  The rate of how fast the signal can change.
- *Running Variance*
  How much a data point can vary from the nearby data points.

For instance, the cooling fluid of the car can increase by 10 °C per minute(high) and decrease by -5 °C per minute(low), but never +/- 50 °C per minute (minimum and maximum).

Hence, the confidence in each parameter is obtained and the overall score is simply the minimum of each of the four separate parameter scores. This gives a fast indication of the confidence that one can have in the data point. To analyse a sensor over time a histogram can then be used including how many points below a threshold (e.g., 75) is accepted in a given time period.

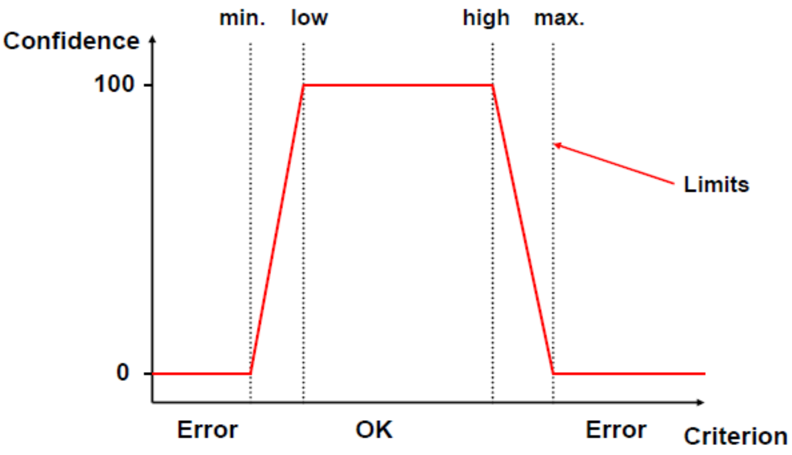


Figure 28: Confidence level calculation.

6.4.4 Benford's Law testing

Benford's law is typically used to detect fraud in systems. When a series of numbers occur, the first digit follows the distribution described by Benford (lower numbers are more frequent than higher numbers).

For data validation, Benford's law can be used to find errors in the system, or to evaluate if upgrades has had a significant improvement into the validity of the dataset.

In the Paper ”On the Use of Benford's Law to Assess the Quality of the Data Provided by Lightning Locating Systems” by E. Mansouri, 2022, the data followed Benford's law better and better as the lightning detection systems were upgraded. Hence, was able to give an estimation on the improvement of the lightning strokes detected by the system.

6.4.5 Bias and outlier detection for a group of sensors

Investigating if certain sensors are outlying or have an offset in comparison to other sensors that should have similar data is a good way of validating data. The best way to do that is to calculate the mean of all the sensors and subtract that value from the data from each sensor. Afterwards it is possible to evaluate if there are deviations from the mean and the spreading of the sensors in a given time period. In Figure 29 first all data from temperature sensors in parking sensors embedded in the tarmac below parking spots is plotted. Having corrected for the mean, it is possible to see that the temperature variations from March to September are very large across the sensors. But in the time period November and February it is very stable. The purpose of using the data was to detect if salting of the roads to avoid freezing was needed. It was determined possible after the evaluation. Additionally, a few sensors (brown and orange curves) were seen to have a negative offset in comparison to the rest of the sensors.

It should be mentioned that the large variations were primarily caused by solar radiation onto the parking sensors, which lead to a higher temperature reading than the ambient air temperature.

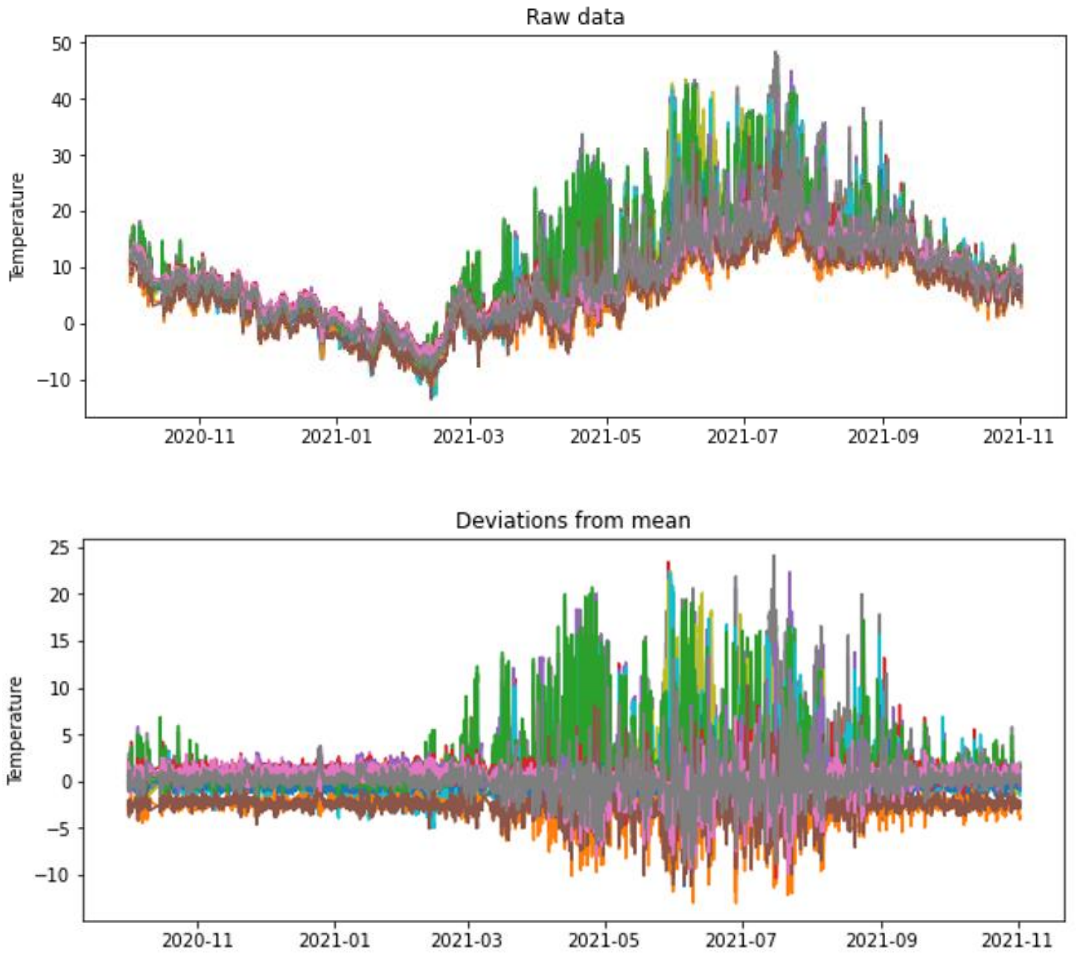
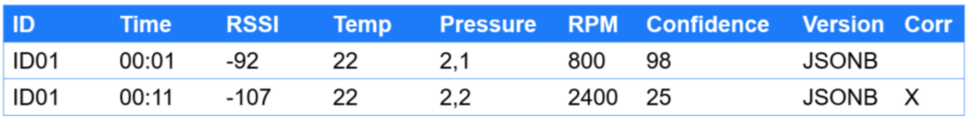


Figure 29: Data from temperature sensors in parking sensors embedded in the tarmac below parking spots. Top: Raw data, Bottom: Raw data with the mean value of all sensors in that time subtracted.

6.4.6 Data correction algorithms

An essential aspect of data validation is also data versioning. The ideal scenario is to have two separate data bases with the same data, except that the first contains raw data, and the second contains the cleaned data. However, in practice, this would in some cases lead to a large extra need for storage capacity. As data needs to be corrected, for instance to correct for the off-set shown in Figure 29, or additional data associated with the data point, such as the confidence level thresholds – see section 0, the update history of the data should be documented.

The version history can be an object, that contains the entire previous data line, a reference to the algorithm that corrected it, and when it was corrected. The corrected flag can also be embedded in the version history object and/or contain richer information such as the type of correction(s) applied.



ID	Time	RSSI	Temp	Pressure	RPM	Confidence	Version	Corr
ID01	00:01	-92	22	2,1	800	98	JSONB	
ID01	00:11	-107	22	2,2	2400	25	JSONB	X

Figure 29: Version history object added in a database.

Data corrections can be divided into two categories for IoT systems, metadata and timeseries data.

Metadata is typically user inputs or data imported from other systems. Here corrections are

- *Format changes*
  Changes from one type of format to another – e.g., date format.
- *Data alignment*
  A reference dataset is used to find the right values, such as typing errors in street names.
- *Schema alignment*
  Data is aligned to the schema, e.g., to not allow special characters in street names.

After evaluations of the data, such as with the confidence level calculations, warnings and errors has been flagged in the dataset. Based on these evaluations, new values can be inserted in the time series data (while tracking the change history). This can be based on several rules:

- *Scale regularization*
  Conversion of values in range from 0 to 1 – to align with 0-100%.
- *Unit change*
  Conversion of value from one unit to another – e.g., battery voltage or percent remaining.
- *Value insertion*
  If the value is in error range e.g., missing due to loss of signal or physically impossible multiple options exist.
- *Interpolation*
  Value from Interpolation between the two adjacent values or based on linear regression on previous points – can be based on more advanced algorithms such as local polynomial.
- *Smoothing*
  A specific variant of interpolation where data is smoothed for anomalies.
- *Inference calculation*
  Value calculated from related measurements based on the physical model of the system, e.g., water level in tank calculated from rain fall measurements.
- *Calibration correction*
  Value correction after a sensor has been calibrated wrt. gain and offset.
- *External data insertion*
  Insertion of data points from an external source, could be meteorological data instead of rain sensor data.

The applied rules will differentiate from case to case, but “change history” should always be documented.

6.5 Calibrations

Calibrations are essential to validate data in an IoT system. A calibration determines how accurate the measured sensor value is compared to the true physical value. However, please recall the notes in section 0 about installation validation.

If a sensor is not trusted with respect to accuracy, it can be (re)-calibrated and the validity of the data from the source can be used. Calibrations fall into two primary categories:

- *Lab calibrations*: Performed in an accredited lab.
- *Field or in-situ calibrations*: Performed at the site of the installation.

6.5.1 Lab calibrations

Lab calibrations can either be performed in house or at an external lab. The important feature is that the calibration relates the sensor measurements to a known traceable reference. This may seem very expensive, and sometimes it can be, but can also be performed rather cheaply. For instance, by having one calibrated temperature sensor, and then a procedure to compare that sensor to all the other sensors. ISO 17025 has guidance for what is required to do that. Outcomes of a calibration should be

- Measurement uncertainty
- Measurement linearity
- Valid range of measurements

In some cases, also the time response, i.e., the time it takes before a measurement is stable when the environment has changed, and other parameters can also be outcomes of a calibration.

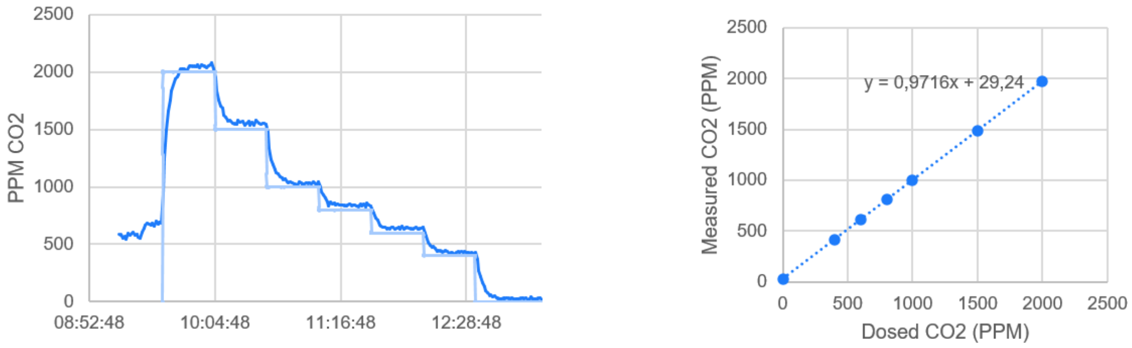


Figure 30: Lab calibration results of a CO2 sensor. Left: Light blue: Exposure reference, Dark blue: Measured value, Right: Linearity of sensor.

Almost any physical quantity can be calibrated. It is more a matter of finding the lab, which can do that, and considering how to cost optimize, since often the sensor calibration can cost the same as the sensor itself.

6.5.2 In-situ calibrations

In-situ calibrations are not performed in the lab, but at the location of the sensor. Here the known reference is brought to the sensor and keeps a parallel data source, which is traceable to a known reference, to the sensor that needs to be calibrated. These are most often used as recalibrations.

Particularly for IoT calibrations, the data in the backend during the calibration can also serve as a reference for the validity of other sensors in the vicinity of that sensor. It will keep performing in-situ calibrations for the other un-calibrated sensors. A good example of this could be to have five known calibrated sensors in the data shown in Figure 29.

6.5.3 Sensor fusion

Sensor fusion is referred to having other sensors measuring other, or the same, physical quantities than the original sensor, to improve the data for that original sensor. This can be an advantage to validate the data. For instance, having a solar irradiation sensor next to the temperature sensors in Figure 29, can create a secondary data trace that can determine, if the solar irradiation at that sensor makes the measurement invalid. Similarly, the parking sensor, which is mounted in the same sensor, would indicate that a car is parked on top of the sensor, and therefore, will not be hit by the sun. That could also be used to improve the validity of the temperature sensor data.

7 Conclusion

The present whitepaper has described the best practices for data validation. It is evident that data validation can quickly become a very time-consuming task and therefore it is important to reiterate, that the most essential activity of data validation is the definition of the requirements that data must meet to be fit-for-purpose. Is the quality and accuracy sufficient to be used as the data for the data driven decisions? Having determined this and adopting a very critical prioritization of activities in the iterative process can improve the validity and quality of your data substantially with a minimum effort.